

	MACROPROCESO DE APOYO	CÓDIGO: AAAr113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
		PAGINA: 1 de 8

26.

FECHA	martes, 29 de mayo de 2018
--------------	----------------------------

Señores
UNIVERSIDAD DE CUNDINAMARCA
 BIBLIOTECA
 Ciudad

UNIDAD REGIONAL	Extensión Facativá
TIPO DE DOCUMENTO	Trabajo De Grado
FACULTAD	Ingeniería
NIVEL ACADÉMICO DE FORMACIÓN O PROCESO	Pregrado
PROGRAMA ACADÉMICO	Ingeniería de Sistemas

El Autor(Es):

APELLIDOS COMPLETOS	NOMBRES COMPLETOS	No. DOCUMENTO DE IDENTIFICACIÓN
Castellanos Rodríguez	Héctor Hernán	1'073.511.416
Osorio Franco	John Fredy	1'070.978.963

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
 Teléfono (091) 8281483 Línea Gratuita 018000976000
 www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
 NIT: 890.680.062-2

*Documento controlado por el Sistema de Gestión de la Calidad
 Asegúrese que corresponde a la última versión consultando el Portal Institucional*

	MACROPROCESO DE APOYO	CÓDIGO: AAAr113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
		PAGINA: 2 de 8

Director(Es) y/o Asesor(Es) del documento:

APELLIDOS COMPLETOS	NOMBRES COMPLETOS
Barahona Rodríguez	Cesar Yesid

TÍTULO DEL DOCUMENTO
MÓDULO PARA LA EVALUACIÓN DE ESTÁNDARES DE CODIFICACIÓN BAJO LA METODOLOGÍA DE CALIDAD DE SOFTWARE PARA LA UNIVERSIDAD DE CUNDINAMARCA

SUBTÍTULO (Aplica solo para Tesis, Artículos Científicos, Disertaciones, Objetos Virtuales de Aprendizaje)

TRABAJO PARA OPTAR AL TÍTULO DE: Aplica para Tesis/Trabajo de Grado/Pasantía
Ingeniero de Sistemas

AÑO DE EDICIÓN DEL DOCUMENTO	NÚMERO DE PÁGINAS
29/05/2018	233

DESCRIPTORES O PALABRAS CLAVES EN ESPAÑOL E INGLÉS (Usar 6 descriptores o palabras claves)	
ESPAÑOL	INGLÉS
1. Estándares de codificación	Coding Standards
2. Calidad de Software	Software Quality
3. Mantenibilidad	Maintainability
4. Integridad de Código	Code Integrity
5. Productos de Software	Software Products
6.	

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
Teléfono (091) 8281483 Línea Gratuita 018000976000
www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
NIT: 890.680.062-2

*Documento controlado por el Sistema de Gestión de la Calidad
Asegúrese que corresponde a la última versión consultando el Portal Institucional*



MACROPROCESO DE APOYO	CÓDIGO: AAAr113
PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
	PAGINA: 3 de 8

RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS

(Máximo 250 palabras – 1530 caracteres, aplica para resumen en español):

Resumen

La universidad de Cundinamarca cuenta con un aplicativo web llamado Calisoft, con el cual se realiza la evaluación de calidad de los productos de software desarrollados por los estudiantes. Este aplicativo permite realizar pruebas de funcionalidad, evaluar el modelado del sistema desarrollado, entre otras funcionalidades.

En este proyecto se busca desarrollar un módulo para este aplicativo, el cual permita evaluar el estándar de codificación en los ficheros de código fuente entregados por los estudiantes. Para esto primero se definió un estándar de codificación que fuera conocido y además estuviera acorde a las tecnologías que actualmente se implementan en los desarrollos de la universidad.

La forma de evaluar este estándar fue parte fundamental de este trabajo, para eso se elaboró una métrica de evaluación la cual define una serie de directrices que están acorde a las recomendaciones del estándar seleccionado. A cada una de estas directrices se le asignó una prioridad de acuerdo con la frecuencia de uso y de su importancia en el código fuente.

El software desarrollado trabaja sobre los ficheros de código fuente, los cuales analiza, para encontrar cada uno de los elementos que se van a evaluar, después de esto determina si cada uno de estos, cumple con las normas establecidas por el estándar para aprobarlos y reprobarlos. Por último, el módulo genera una calificación de acuerdo a la métrica elaborada

Abstract

The University of Cundinamarca has a web application called Calisoft, with which the quality assessment of the software products developed by the students. This application allows to perform functionality tests, evaluate the modeling of the developed system, among other functionalities.

This project seeks to develop a module for this application, which allows to evaluate the coding standard in the source code files delivered by the students. For this, a coding standard was first defined that was known and was also in line with the technologies currently implemented in the university's developments.

The way to evaluate this standard was a fundamental part of this work, for which an evaluation metric was developed which defines a series of guidelines that are in accordance with the recommendations of the selected standard. Each of these guidelines was assigned a priority according to the frequency of use and its importance in the source code.

The developed software works on the files of source code, which analyzes, to find each one of the elements that are going to be evaluated, after this it determines if each one of these, complies with the norms established by the standard to approve and reject them. Finally, the module generates a rating according to the metric developed.



MACROPROCESO DE APOYO	CÓDIGO: AAAR113
PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
	PAGINA: 4 de 8

AUTORIZACION DE PUBLICACIÓN

Por medio del presente escrito autorizo (Autorizamos) a la Universidad de Cundinamarca para que, en desarrollo de la presente licencia de uso parcial, pueda ejercer sobre mí (nuestra) obra las atribuciones que se indican a continuación, teniendo en cuenta que, en cualquier caso, la finalidad perseguida será facilitar, difundir y promover el aprendizaje, la enseñanza y la investigación.

En consecuencia, las atribuciones de usos temporales y parciales que por virtud de la presente licencia se autoriza a la Universidad de Cundinamarca, a los usuarios de la Biblioteca de la Universidad; así como a los usuarios de las redes, bases de datos y demás sitios web con los que la Universidad tenga perfeccionado una alianza, son: Marque con una "X":

AUTORIZO (AUTORIZAMOS)	SI	NO
1. La reproducción por cualquier formato conocido o por conocer.	X	
2. La comunicación pública por cualquier procedimiento o medio físico o electrónico, así como su puesta a disposición en Internet.	X	
3. La inclusión en bases de datos y en sitios web sean éstos onerosos o gratuitos, existiendo con ellos previa alianza perfeccionada con la Universidad de Cundinamarca para efectos de satisfacer los fines previstos. En este evento, tales sitios y sus usuarios tendrán las mismas facultades que las aquí concedidas con las mismas limitaciones y condiciones.	X	
4. La inclusión en el Repositorio Institucional.	X	

De acuerdo con la naturaleza del uso concedido, la presente licencia parcial se otorga a título gratuito por el máximo tiempo legal colombiano, con el propósito de que en dicho lapso mi (nuestra) obra sea explotada en las condiciones aquí estipuladas y para los fines indicados, respetando siempre la titularidad de los derechos patrimoniales y morales correspondientes, de acuerdo con los usos honrados, de manera proporcional y justificada a la finalidad perseguida, sin ánimo de lucro ni de comercialización.

Para el caso de las Tesis, Trabajo de Grado o Pasantía, de manera complementaria, garantizo(garantizamos) en mi(nuestra) calidad de estudiante(s) y por ende autor(es) exclusivo(s), que la Tesis, Trabajo de Grado o Pasantía en cuestión, es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi(nuestra) creación original particular y, por tanto, soy(somos)



MACROPROCESO DE APOYO	CÓDIGO: AAAr113
PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
	PAGINA: 5 de 8

el(los) único(s) titular(es) de la misma. Además, aseguro (aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos de la Tesis o Trabajo de Grado es de mí (nuestra) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

Sin perjuicio de los usos y atribuciones otorgadas en virtud de este documento, continuaré (continuaremos) conservando los correspondientes derechos patrimoniales sin modificación o restricción alguna, puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación de los derechos patrimoniales derivados del régimen del Derecho de Autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “*Los derechos morales sobre el trabajo son propiedad de los autores*”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables. En consecuencia, la Universidad de Cundinamarca está en la obligación de RESPETARLOS Y HACERLOS RESPETAR, para lo cual tomará las medidas correspondientes para garantizar su observancia.

NOTA: (Para Tesis, Trabajo de Grado o Pasantía):

Información Confidencial:

Esta Tesis, Trabajo de Grado o Pasantía, contiene información privilegiada, estratégica, secreta, confidencial y demás similar, o hace parte de la investigación que se adelanta y cuyos resultados finales no se han publicado.

SI ___ NO X .

En caso afirmativo expresamente indicaré (indicaremos), en carta adjunta tal situación con el fin de que se mantenga la restricción de acceso.

LICENCIA DE PUBLICACIÓN

Como titular(es) del derecho de autor, confiero(erimos) a la Universidad de Cundinamarca una licencia no exclusiva, limitada y gratuita sobre la obra que se integrará en el Repositorio Institucional, que se ajusta a las siguientes características:

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
Teléfono (091) 8281483 Línea Gratuita 018000976000
www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
NIT: 890.680.062-2



MACROPROCESO DE APOYO	CÓDIGO: AAAR113
PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
	PAGINA: 6 de 8

- a) Estará vigente a partir de la fecha de inclusión en el repositorio, por un plazo de 5 años, que serán prorrogables indefinidamente por el tiempo que dure el derecho patrimonial del autor. El autor podrá dar por terminada la licencia solicitándolo a la Universidad por escrito. (Para el caso de los Recursos Educativos Digitales, la Licencia de Publicación será permanente).
- b) Autoriza a la Universidad de Cundinamarca a publicar la obra en formato y/o soporte digital, conociendo que, dado que se publica en Internet, por este hecho circula con un alcance mundial.
- c) Los titulares aceptan que la autorización se hace a título gratuito, por lo tanto, renuncian a recibir beneficio alguno por la publicación, distribución, comunicación pública y cualquier otro uso que se haga en los términos de la presente licencia y de la licencia de uso con que se publica.
- d) El(Los) Autor(es), garantizo(amos) que el documento en cuestión, es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi (nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro(aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos es de mí (nuestro) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.
- e) En todo caso la Universidad de Cundinamarca se compromete a indicar siempre la autoría incluyendo el nombre del autor y la fecha de publicación.
- f) Los titulares autorizan a la Universidad para incluir la obra en los índices y buscadores que estimen necesarios para promover su difusión.
- g) Los titulares aceptan que la Universidad de Cundinamarca pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.
- h) Los titulares autorizan que la obra sea puesta a disposición del público en los términos autorizados en los literales anteriores bajo los límites definidos por la universidad en el “Manual del Repositorio Institucional AAAM003”
- i) Para el caso de los Recursos Educativos Digitales producidos por la Oficina de Educación Virtual, sus contenidos de publicación se rigen bajo la Licencia Creative Commons: Atribución- No comercial- Compartir Igual.

	MACROPROCESO DE APOYO	CÓDIGO: AAAr113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
		PAGINA: 7 de 8



j) Para el caso de los Artículos Científicos y Revistas, sus contenidos se rigen bajo la Licencia Creative Commons Atribución- No comercial- Sin derivar.



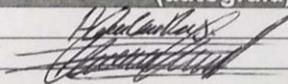
Nota:

Si el documento se basa en un trabajo que ha sido patrocinado o apoyado por una entidad, con excepción de Universidad de Cundinamarca, los autores garantizan que se ha cumplido con los derechos y obligaciones requeridos por el respectivo contrato o acuerdo.

La obra que se integrará en el Repositorio Institucional, está en el(los) siguiente(s) archivo(s).

Nombre completo del Archivo Incluida su Extensión (Ej. PerezJuan2017.pdf)	Tipo de documento (ej. Texto, imagen, video, etc.)
1. Libro final del modulo de codificacion.pdf	TEXTO, IMAGENES
2. Manual de usuario Codificación.pdf	TEXTO, IMÁGENES
3. manual de instalación.pdf	TEXTO, IMÁGENES
4. Codigo de codificación.rar	software
5. Articulo EC en CITTI.pdf	TEXTO, IMÁGENES
6. Articulo EC Colcaci.pdf	TEXTO, IMAGENES

En constancia de lo anterior, Firmo (amos) el presente documento:

APELLIDOS Y NOMBRES COMPLETOS	FIRMA (autógrafa)
Castellanos Rodríguez Herta Hernán Osorio Franco John Fredy	

12.1.50



MACROPROCESO DE APOYO	CÓDIGO: AAAr113
PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
	PAGINA: 8 de 8

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
Teléfono (091) 8281483 Línea Gratuita 018000976000
www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
NIT: 890.680.062-2

*Documento controlado por el Sistema de Gestión de la Calidad
Asegúrese que corresponde a la última versión consultando el Portal Institucional*

**MODULO PARA LA EVALUACIÓN DE ESTÁNDARES DE CODIFICACIÓN
BAJO LA METODOLOGÍA DE CALIDAD DE SOFTWARE PARA LA
UNIVERSIDAD DE CUNDINAMARCA**

**HÉCTOR HERNÁN CASTELLANOS RODRÍGUEZ
JOHN FREDY OSORIO FRANCO**

**UNIVERSIDAD DE CUNDINAMARCA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
FACATATIVÁ**

2018

**MODULO PARA LA EVALUACIÓN DE ESTÁNDARES DE CODIFICACIÓN
BAJO LA METODOLOGÍA DE CALIDAD DE SOFTWARE PARA LA
UNIVERSIDAD DE CUNDINAMARCA**

**HÉCTOR HERNÁN CASTELLANOS RODRÍGUEZ
JOHN FREDY OSORIO FRANCO**

**Director: Ms. Barahona Rodríguez Cesar Yesid, Especialista en
Gestión de Proyectos**

**GRUPO DE INVESTIGACIÓN DE SISTEMAS Y TECNOLOGÍA DE
FACATATIVÁ (GISTFA)**

**UNIVERSIDAD DE CUNDINAMARCA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
FACATATIVÁ**

2018

Nota de aceptación

Firma del jurado

Firma del jurado

Facatativá, 02 de Mayo del 2018

Los autores queremos dedicar este trabajo de grado principalmente a nuestras familias por su apoyo y constante colaboración en nuestro proceso de formación como personas y profesionales, gracias a esto hoy nos permite culminar con éxito una meta importante en nuestras vidas.

AGRADECIMIENTOS

Los autores queremos agradecer primeramente a nuestras familias, que nos han apoyado durante todo nuestro proceso de formación como personas y profesionales, gracias a ellos podemos culminar con éxito esta etapa importante de nuestras vidas.

De igual forma queremos agradecer a Dios que nos ha brindado la fuerza y la salud para llevar a buen término este proyecto.

Asimismo, agradecemos a nuestro director de tesis Ms. Cesar Yesid Barahona Rodríguez, quien nos ha orientado y apoyado siempre de la mejor forma, en cada una de las etapas de este proyecto que hoy terminamos.

Gracias a cada uno de los docentes, que durante este tiempo nos han brindado su conocimiento, enseñanzas y experiencias, estas nos han servido como motor de inspiración para seguir adelante en este camino que hace años decidimos comenzar con ilusión y alegría.

COMPROMISO DE AUTOR

Yo, Héctor Hernán Castellanos Rodríguez con cedula de identidad No 1'073.511.416 y con código No 461.214.108 estudiante de ingeniería de sistemas de la universidad de Cundinamarca, declaro que:

El contenido del presente documento es un reflejo de mi trabajo personal y manifiesto que, ante cualquier notificación de plagio, copia o falta a la fuente original, soy responsable directo legal, económico y administrativo sin afectar al Director del trabajo, a la Universidad y a cuantas instituciones hayan colaborado en dicho trabajo, asumiendo las consecuencias derivadas de tales prácticas.

FIRMA:

COMPROMISO DE AUTOR

Yo, John Fredy Osorio Franco con cedula de identidad No 1'070.978.963 y con código No 461.214.146 estudiante de ingeniería de sistemas de la universidad de Cundinamarca, declaro que:

El contenido del presente documento es un reflejo de mi trabajo personal y manifiesto que, ante cualquier notificación de plagio, copia o falta a la fuente original, soy responsable directo legal, económico y administrativo sin afectar al Director del trabajo, a la Universidad y a cuantas instituciones hayan colaborado en dicho trabajo, asumiendo las consecuencias derivadas de tales prácticas.

FIRMA:

RESUMEN

La universidad de Cundinamarca cuenta con un aplicativo web llamado Calisoft, con el cual se realiza la evaluación de calidad de los productos de software desarrollados por los estudiantes. Este aplicativo permite realizar pruebas de funcionalidad, evaluar el modelado del sistema desarrollado, entre otras funcionalidades.

En este proyecto se busca desarrollar un módulo para este aplicativo, el cual permita evaluar el estándar de codificación en los ficheros de código fuente entregados por los estudiantes. Para esto primero se definió un estándar de codificación que fuera conocido y además estuviera acorde a las tecnologías que actualmente se implementan en los desarrollos de la universidad.

La forma de evaluar este estándar fue parte fundamental de este trabajo, para eso se elaboró una métrica de evaluación la cual define una serie de directrices que están acorde a las recomendaciones del estándar seleccionado. A cada una de estas directrices se le asignó una prioridad de acuerdo con la frecuencia de uso y de su importancia en el código fuente.

El software desarrollado trabaja sobre los ficheros de código fuente, los cuales analiza, para encontrar cada uno de los elementos que se van a evaluar, después de esto determina si cada uno de estos, cumple con las normas establecidas por el estándar para aprobarlos y reprobarlos. Por último, el módulo genera una calificación de acuerdo a la métrica elaborada.

ABSTRACT

The University of Cundinamarca has a web application called Calisoft, with which the quality assessment of the software products developed by the students. This application allows to perform functionality tests, evaluate the modeling of the developed system, among other functionalities.

This project seeks to develop a module for this application, which allows to evaluate the coding standard in the source code files delivered by the students. For this, a coding standard was first defined that was known and was also in line with the technologies currently implemented in the university's developments.

The way to evaluate this standard was a fundamental part of this work, for which an evaluation metric was developed which defines a series of guidelines that are in accordance with the recommendations of the selected standard. Each of these guidelines was assigned a priority according to the frequency of use and its importance in the source code.

The developed software works on the files of source code, which analyzes, to find each one of the elements that are going to be evaluated, after this it determines if each one of these, complies with the norms established by the standard to approve and reject them . Finally, the module generates a rating according to the metric developed.

Tabla de contenido

1. INTRODUCCION	18
2. INFORME DE INVESTIGACION	19
2.1 Estado del arte	19
2.2 Línea de investigación.....	26
2.3 Planteamiento del problema.....	26
2.4 Objetivos	27
2.4.1 Objetivo general.....	27
2.4.2 Objetivos específicos	27
2.5 Impacto del proyecto	27
2.6 Metodología.....	28
2.7 Marcos de referencia.....	28
2.7.1 Marco Histórico.....	28
2.7.2 Marco teórico	31
2.7.3 Marco legal	33
3. DOCUMENTACION DEL SOFTWARE.....	35
3.1 Plan de proyecto	35
3.2 Determinación de requerimientos.....	36
3.2.1 Requerimientos funcionales.....	36
3.2.2 Definición de requerimientos funcionales	39
3.2.3 Requerimientos no funcionales.....	41
3.2.4 Requisitos funcionales de las interfaces	43
4. Especificación del sistema	44
4.1 Estándar de codificación	44
4.1.1 Variables.....	44
4.1.2 Clases.....	44
4.1.3 Funciones	45
4.1.4 Constantes.....	45
4.1.5 Indentacion	46
4.1.6 Comentarios	47
4.1.7 Espacios de nombre	48

4.2 Métrica para la evaluación de estándares de codificación	49
4.2.1 Ecuaciones	50
5. Especificación de diseño	54
5.1 Modelo entidad relación	54
5.1.1 Descripción modelo entidad relación del modulo.....	56
5.2 Diagramas casos de uso	57
5.2.1 Estudiante.....	57
5.2.2 Evaluador.....	59
5.2.3 Administrador.....	61
5.3 Diagramas de secuencia.....	62
5.4 Diagramas de actividades	82
5.5 Diagrama de clases.....	99
6. CASOS DE PRUEBA.....	101
6.1 Diseño casos de prueba.....	101
7. ESTIMACION DE RECURSOS	105
7.1 Resumen por rubros.....	105
7.2 Detalle personal	105
7.3 Descripción de equipos	106
7.4 Descripción de materiales e insumos.....	106
7.5 Descripción de servicios tecnológicos.....	106
7.6 Descripción de viajes	107
7.7 Descripción de otros.....	107
8. RESULTADOS.....	108
8.1 Evaluación declaración de variables	108
8.2 Evaluación declaración de funciones	110
8.3 Evaluación declaración de clases	111
8.4 Evaluación indentacion estructuras de control	113
8.5 Evaluación uso de comentarios en el código fuente	114
8.6 Evaluación declaración de constantes	115
8.7 Evaluación declaración de espacios de nombre	117
9. CONCLUSIONES Y RECOMENDACIONES	119
REFERENCIAS	120
ANEXOS.....	123

Lista de tablas

Tabla 1. Prioridad de cada directriz.	50
Tabla 2. Descripción diagrama entidad relación.	57
Tabla 3. Descripción caso de uso subir script.....	58
Tabla 4. Descripción caso de eliminar script.....	58
Tabla 5. Descripción caso de uso listar script.....	58
Tabla 6. Descripción caso de uso ver evaluación modulo.	59
Tabla 7. Descripción caso de uso seleccionar proyecto a evaluar.....	59
Tabla 8. Descripción caso de uso evaluar estándar.	60
Tabla 9. Descripción caso de uso visualizar código.....	60
Tabla 10. Descripción caso de uso comentar.	60
Tabla 11. Descripción caso de uso generar calificación.	60
Tabla 12. Descripción caso de uso ver reporte de calificación modulo.....	61
Tabla 13. Descripción caso de uso ver reporte de calificación general.	62
Tabla 14. Descripción caso de uso cambiar criterio de evaluación.....	62
Tabla 15. Descripción diagrama de secuencia subir scripts.	64
Tabla 16. Descripción diagrama de secuencia ver evaluación.	66
Tabla 17. Descripción diagrama de secuencia proyecto a evaluar.....	68
Tabla 18. Descripción diagrama de secuencia evaluar estándar.....	70
Tabla 19. Descripción diagrama de secuencia visualizar código fuente.	73
Tabla 20. Descripción diagrama de secuencia generar calificación.....	75
Tabla 21. Descripción diagrama de secuencia ver reporte de calificación....	79
Tabla 22. Descripción diagrama de secuencia ver reporte general	80
Tabla 23. Descripción diagrama de secuencia cambio criterios	82
Tabla 24. Descripción diagrama de actividades subir script.	84
Tabla 25. Descripción diagrama de actividades ver evaluación del módulo. 85	
Tabla 26. Descripción diagrama de actividades seleccionar proyecto.....	86
Tabla 27. Diagrama de actividades escoger script de codificación.....	87
Tabla 28. Descripción diagrama de actividades seleccionar script.....	89
Tabla 29. Descripción diagrama de actividades visualizar código fuente.	91
Tabla 30. Descripción diagrama de actividades generar calificación.....	93
Tabla 31. Descripción diagrama de actividades generar comentario.....	95
Tabla 32. Descripción diagrama de actividades generar reporte evaluador. 96	
Tabla 33. Descripción diagrama de actividades cambiar criterio.	98
Tabla 34. Descripción diagrama de clases.	100
Tabla 35. Caso prueba declaración de variables.	101
Tabla 36. Caso prueba declaración de funciones.	101
Tabla 37. Caso prueba declaración de clases.	102
Tabla 38. Caso prueba indentacion estructuras de control.....	102
Tabla 39. Caso de prueba uso de comentarios en el código fuente.	103
Tabla 40. Caso prueba declaración de constantes.	103
Tabla 41. Caso prueba declaración de espacios de nombre.	104

Tabla 42. Resumen por rubros.	105
Tabla 43. Detalles personales.....	105
Tabla 44. Descripción de equipos.....	106
Tabla 45. Descripción de materiales e insumos.....	106
Tabla 46. Descripción de servicios tecnológicos.....	106
Tabla 47. Descripción de viajes.	107
Tabla 48. Descripción de otros.	107

Lista de figuras

Figura 1. Esquema de Calificación de Rees.	21
Figura 2. Visión general del proceso de medición	24
Figura 3. Cronograma de actividades.	35
Figura 4. Ejemplo declaración de variables.	44
Figura 5. Ejemplo declaración de clases.	45
Figura 6. Ejemplo declaración de funciones.	45
Figura 7. Ejemplo declaración de constantes.	46
Figura 8. Forma correcta de indentacion.	47
Figura 9. Forma incorrecta de indentacion.	47
Figura 10. Uso de comentarios en el código fuente.	48
Figura 11. Ejemplo declaración de espacios de nombre.	48
Figura 12. Modelo entidad relación general.	55
Figura 13. Segmento de diagrama para el módulo de estándares	55
Figura 14. Caso de uso subir script.	57
Figura 15. Caso de uso ver evaluación modulo.	58
Figura 16. Caso de uso seleccionar proyecto a evaluar.	59
Figura 17. Caso de uso ver reporte calificación modulo.	61
Figura 18. Caso de uso ver reporte de calificación general.	61
Figura 19. Caso de uso cambiar criterio de evaluación.	62
Figura 20. Diagrama de secuencia subir scripts.	63
Figura 21. Diagrama de secuencia ver evaluación.	65
Figura 22. Diagrama de secuencia proyecto a evaluar.	67
Figura 23. Diagrama de secuencia evaluar estándar.	69
Figura 24. Diagrama de secuencia visualizar código fuente.	71
Figura 25. Diagrama de secuencia generar calificación.	73
Figura 26. Diagrama de secuencia ver reporte de calificación.	76
Figura 27. Diagrama de secuencia ver reporte general del módulo.	79
Figura 28. Diagrama de secuencia cambio criterios de evaluación.	81
Figura 29. Diagrama de actividades subir script.	83
Figura 30. Diagrama de actividades ver evaluación del módulo.	85
Figura 31. Diagrama de actividades seleccionar proyecto a evaluar.	86
Figura 32. Diagrama de actividades escoger script de codificación.	87
Figura 33. Diagrama de actividades seleccionar script.	88
Figura 34. Diagrama de actividades visualizar código fuente.	90
Figura 35. Diagrama de actividades generar calificación.	92
Figura 36. Diagrama de actividades generar comentario.	94
Figura 37. Diagrama de actividades generar reporte evaluador.	96
Figura 38. Diagrama de actividades cambiar criterios de evaluación.	97
Figura 39. Diagrama de clases.	99
Figura 40. Fichero declaración de variables.	109
Figura 41. Reporte evaluación fichero variables.php.	109
Figura 42. Fichero declaración de funciones.	110

Figura 43. Reporte evaluación fichero funciones.php.....	111
Figura 44. Fichero declaración de clases.	112
Figura 45. Reporte evaluación fichero clases.php.	112
Figura 46. Fichero estructuras de control.	113
Figura 47. Reporte de evaluación fichero indentacion.php.	113
Figura 48. Fichero uso de comentarios en el código fuente.	114
Figura 49. Reporte de evaluación fichero comentarios.php.....	115
Figura 50. Fichero declaración de constantes.	116
Figura 51. Reporte evaluación fichero constantes.php.....	116
Figura 52. Fichero declaración espacios de nombre.	117
Figura 53. Reporte de evaluación fichero espacios de nombre.php	117

Lista de ecuaciones

Ecuación 1. Calificación de variables.....	50
Ecuación 2. Calificación de métodos.	51
Ecuación 3. Calificación de clases.....	51
Ecuación 4. Calificación de indentacion.....	51
Ecuación 5. Calificación de comentarios.	52
Ecuación 6. Calificacion de constantes.....	52
Ecuación 7. Calificacion de espacios de nombre.....	52
Ecuación 8. Cálculo del índice de cohesión.....	53

Lista de anexos

Anexo A. Manual de usuario	123
Anexo B. Manual de instalación.....	166
Anexo C. Articulo ponencia universidad san buenaventura.....	204
Anexo D. Articulo transferencia colcaci 2018.....	215
Anexo E. Registro derechos de autor	224
Anexo F. Reportes pruebas de calidad.....	225

1. INTRODUCCION

El uso de estándares de codificación, es una práctica altamente recomendada para desarrollar software de alta calidad, los estándares permiten establecer criterios únicos que los programadores deben implementar cuando escriben código, al utilizar un estándar de codificación se busca que el código fuente pueda ser entendido por cualquier miembro del equipo de desarrollo, esto permite que el código pueda ser modificado por otro programador evitando así la dependencia de personal o en el peor de los casos tener que volver a escribir la totalidad del código, lo que ocasionaría costos extras y mayor tiempo del requerido.

Los desarrollos a gran escala pueden estar constituidos por grandes cantidades de archivos y millones de líneas de código, esto en muchas ocasiones impide que se pueda hacer una revisión detallada de cómo está escrito el código y si se implementó el estándar establecido, debido a esto en este proyecto se quiere desarrollar una herramienta que permita automatizar el proceso de revisión de los estándares de codificación en los proyectos desarrollados por los estudiantes de la universidad de Cundinamarca.

El desarrollo de este proyecto se realizara diseñando una métrica de evaluación que permita calcular en qué nivel se implementó el estándar y poder determinar si cumple con los requerimientos mínimos deseados, en base a esta métrica se desarrolla una herramienta que realice este proceso de evaluación.

2. INFORME DE INVESTIGACION

2.1 Estado del arte

En la actualidad con la alta competitividad que existe en el mercado, es de vital importancia para las compañías el poder garantizar que sus productos sean de la mejor calidad posible; en el área del desarrollo de software este aspecto no pasa desapercibido, el poder asegurar que se está elaborando un producto con los mejores estándares de calidad resulta ser de gran relevancia para las compañías que se dedican al desarrollo de software.

Al hablar de estándares de calidad de software debemos referirnos a la norma **ISO/IEC 25010**, que proporciona una guía para el uso de la nueva serie de estándares internacionales llamada requisitos y evaluación de calidad de productos de software. El modelo de calidad definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que son: adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, portabilidad y mantenibilidad, en este último aspecto pensamos enfocar el desarrollo de este proyecto, entendiendo lo importante que resulta aplicar un estándar de programación a la hora de realizar un código fuente que pueda ser fácilmente interpretado y modificado en el futuro para realizar su correcto mantenimiento (Roa, Morales, & Gutierrez, 2015).

La realimentación hacia una evaluación continua y automática a través de las prácticas de programación proporciona información inmediata al alumno de los errores cometidos en prácticas, permitiendo que puedan ser corregidas y reevaluadas repetidas veces, estableciendo un marco de evaluación donde el alumno pueda aumentar su motivación y decidiendo que nota desea obtener. Estos son los siguientes errores por el cual son susceptibles para ser identificados inmediatamente:

- Incumplimiento de interfaz
- Errores de estilo
- Diseño de codificación
- Errores de funcionalidad
- Errores graves de ejecución
- Insuficiencias en las pruebas

Se emplea una herramienta como soporte para la evaluación automática llamada GAP, este permite la ejecución de herramientas externas de forma segura mediante la simple configuración de líneas de comando del sistema operativo; para la ejecución de las pruebas automáticas del ejercicio basta con escribir un guion usando el lenguaje de *script* del sistema operativo, ejecutando en el momento de la entrega y generando información sobre los

resultados que GAP se encarga de mostrarle a los alumnos a. El profesor puede ejecutar el mismo guion u otro distinto sobre todas las prácticas entregadas si desea obtener información distinta o actualizar la que recibió el alumno en el momento de la entrega operando de la siguiente manera:

Crea un directorio temporal, luego transfiere a ese directorio una copia de los ficheros entregados preparando un *script* para su ejecución en el directorio, seguidamente transfiere el control al *script* y posterior mente elimina el directorio principal. Estableciéndose así un sistema de evaluación automática con herramientas externas.

Se menciona como las pruebas resultan bastantes completas, existiendo determinados aspectos que son difíciles de evaluar de forma automática, se trata de los aspectos cualitativos no cuantificables relacionados como ejemplo se toma las cuestiones de estilo que mide el porcentaje y distribución de los comentarios, y al día de hoy resulta imposible que su contenido sea adecuado, tomando otro ejemplo son los identificadores los cuáles se hacen una pregunta ¿Cómo garantizar que el nombre de la variable este acorde a una función? Tampoco se garantiza que estén utilizando los mecanismos más apropiados del lenguaje en cada situación. En este aspecto, pueden aparecer vicios derivados de las exigencias de las pruebas, como, por ejemplo, que el alumno introduzca comentarios sin sentido para garantizar el porcentaje requerido ejerce así, un impacto visible en los papeles que juegan tanto el profesor como el desarrollador(Rodríguez, Pino, Roca, Figueroa, & González Domínguez, n.d.).

Para asegurar su cooperación y un grado de calidad de software, los programadores profesionales necesitan normas de codificación más seguras, sin embargo, para los programadores principiantes en cuestiones de estilo de programación, se olvidan fácilmente de sus estudios. Entorno a la incorporación de temas de este estilo, los estudiantes aprendices pueden pasar varios cursos sin elementos de aprendizaje de estilo de programación, en esta situación es muy frecuentemente gracias a la escasez de trabajo del tutor para dar a sus estudiantes la información completa sobre sus cursos.

El estilo de programación, es decir, la forma de utilizar un lenguaje de programación y escribir código de programa, es uno de los problemas importantes, pero que a menudo son descuidados en la programación.

Cuando se tiene malos hábitos de escribir mal el código, por ejemplo, no comentar y sin embargo no conocer ningún tipo de estándar de programación, no se podrá entender de manera correcta lo que un desarrollador desea plasmar en su creación de software, tal es el caso que puede haber requisitos de codificación dependientes del lenguaje que requieran atención especial para evitar errores en la funcionalidad un programa.

Tomando como referencia el lenguaje C++, los programadores cuentan con varias fallas como por ejemplo las conversiones implícitas de tipos y métodos de copia, por tal razón si sigue buenas y sistémicas directrices de programación es posible evitar este tipo de problemas, los eventos comunes y normas de codificación son esenciales para la creación de software en la vida real si bien es cierto, los programadores deben ser capaces de trabajar juntos de manera efectiva y prestar atención a la calidad de software.

Una de las primeras obras publicadas de estándares de programación fue una obra publicada por Rees en 1982, quien desarrollo un sistema STYLE para evaluar el estilo de programación en pascal.

Sugiere que al hacer que los temas de estilo fueran visibles y medibles para los estudiantes, aprendieran a prestar atención a ellos. Rees propuso diez mediciones fácilmente calculables. El porcentaje de líneas de comentario, el número de gotos y la longitud media de los identificadores, y creó un esquema de marcado que se basó en cinco parámetros como se ilustra en la Figura 1.

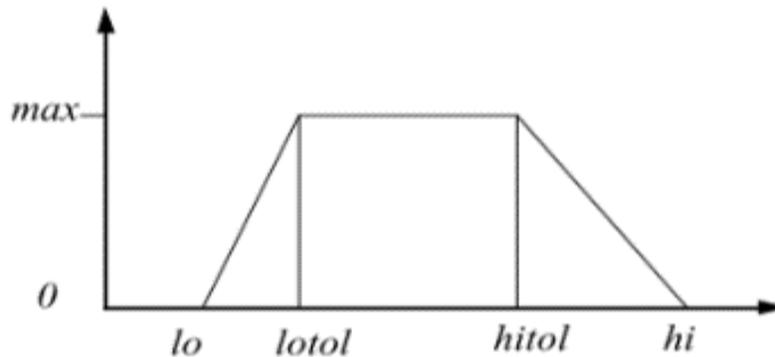


Figura 1. Esquema de Calificación de Rees(Ala-Mutka, Uimonen, & Jarvinen, 2004).

El esquema define un rango para los valores de medición deseados con la marca máxima, Marca cero e interpolación lineal entre estas marcas. Por ejemplo, una cantidad recomendada de líneas de comentario en un programa pequeño puede establecerse en un 20% a 40% de las líneas del programa en total.

Basado en el trabajo de Rees, Berry y Meekings (1985) desarrollaron un analizador de estilo para comprobar el estilo de programación C y probarlo con 80000 líneas de código de programa C. Su trabajo ha sido la base de muchas herramientas de evaluación de estilo de programa C, por ejemplo, Ceilidh (Benford, Burke, & Foxley, 1992) y modificado para Ada en ASSYST (Jackson & Usher, 1997). En estos sistemas, los valores de los parámetros para las

mediciones son generalmente configurables por los profesores del curso(Ala-Mutka et al., 2004).

Los documentos de estándares de programación, define el proceso de codificación como prácticas de matrices hacia una clasificación lo cual se debe adherir los estudiantes al desarrollar código. Existe una norma ISO para los lenguajes de programación en C que permite la implementación por parte de los compiladores y así mismo no todos los compiladores cumplen los estándares de programación, al enseñar a los desarrolladores a diseñar código a un cierto estándar, se promoverán los conceptos de confiabilidad de código, facilidad de mantenimiento y portabilidad, ya que el estudiante tendrá la capacidad de estar preparado para un ambiente de negocios.

“Un estudio demostró empíricamente que la violación de las normas de los estándares, es impacto negativo en un software de fiabilidad”(MacKellar, 2016).

Este estudio utilizo COD, un estándar de sistema embebido, enfocado a la prevención de fallas adoptando reglas de codificación mientras que exista la aplicación de reglas, aprovechando la comprensión de la conformidad de las normas mediante el uso de asignaciones de rosca y la aplicación de las normas. De esta manera el desarrollador vive el proceso en lugar de estudiar la evolución de la industria.

Los estándares de codificación son una parte integral de un software, el autor intento introducir este concepto atreves de la aplicación y la educación mediante el documento de normas de programación. Se proporciona en medio de conferencias integradas con ejemplos adecuadamente, los estándares definidos en el documento y las asignaciones mejoradas para el proceso de desarrollo llamado “código de revisión por pares”. lo cual se centra en la capacidad que tiene el estudiante para innovar dentro de un conjunto de restricciones definidas externamente que tienen como objetivo simular un ambiente de negocio.

El término plagio en el software es difícil de distinguir, Sin embargo, a través del uso de esta norma específicamente en las secciones que tratan con comentarios e identificadores, se hace evidente y más fácil distinguir entre el diseño de problemas de grupo que comparte (que es el aprendizaje) y la copia de código. Además, el uso de los exámenes y revisiones de código de compañeros resaltarán a los estudiantes que no intentan aprender.

El análisis de requisitos para el diseño se captura en el campo de comentario como lo hace la noción de gestión de configuración y plan de prueba de verificación proporcionando al estilo de diseño de código un estándar para la legibilidad, comprensión y facilidad de mantenimiento. Para evaluar el conocimiento de los aprendices sobre el análisis de los requisitos a través del diseño, el uso y la corrección de los comentarios son fundamentales. Esto se

ajusta a las mejores prácticas, Las revisiones de código se utilizan como proceso para reforzar la capacidad de mantenimiento y la conformidad.

Cada función de código fuente () requiere un campo de comentario debido a que el formato que sigue es similar a los estándares que el autor siguió en ocupaciones actuales y anteriores. Estos comentarios que describen la intención de la función (), argumenta en la función, los valores de retorno.

Los estándares indocumentados o ligeramente documentados son normas subjetivas y definen la actitud, el profesionalismo y el trabajo en equipo Finalmente, los comentarios están dirigidos hacia un estilo. Estos son normalmente específicos para el lenguaje de código fuente, ya que son más subjetivos a la evaluación. Como un estándar interno, esto promueve la intención de la aplicación al requisito. Y durante la revisión del código, transferencia de conocimiento al resto del equipo. El estándar identificador, es una variación en la notación CamelCase.

“CamelCase se compone de un número de palabras sin espaciado blanco donde la letra inicial de cada palabra está en mayúscula. CamelCase es una convención de nomenclatura sugerida por Sun Microsystems, comunidades Java y otras comunidades orientadas a objetos.”(MacKellar, 2016).

Una revisión de código es un examen sistemático del código fuente de la computadora. Se pretende verificar que el diseño del código cumpla con los requisitos, encontrar y corregir los errores pasados por alto en la fase de desarrollo inicial, mejorar la calidad general y la capacidad de mantenimiento del software y ayudar en el desarrollo de habilidades y conocimiento del producto por el equipo(MacKellar, 2016).

Durante el paso de los años los estándares de codificación se han vuelto populares como un medio para asegurar la calidad de software durante el proceso de desarrollo de un software, se aseguran un estilo común de programación, lo que aumenta la capacidad de mantenimiento y evitan el uso de construcciones potencialmente generadoras de problemas, aumentando así la confiabilidad. Las reglas en tales estándares se basan principalmente en la opinión experta y ganada por los años con una cierta lengua en varios contextos.

En una investigación reciente sobre las características de los errores. Se argumentó que los primeros controles automatizados han contribuido a la fuerte disminución de los errores de memoria presentes en el software. Sin embargo, la disponibilidad de normas y herramientas adecuadas, existen varias cuestiones hacen obstáculo, Por ejemplo, el 30% de las líneas de uno de los proyectos utilizados en este estudio contenían tal violación. Las infracciones pueden ser subproductos del análisis estático, que no siempre puede determinar si el código viola o no una determinada comprobación.

Tomando otro ejemplo como referencia, encontramos que una sola regla es responsable del 83% de todas las violaciones en uno de los proyectos analizados, lo cual es poco probable que solo señale los verdaderos problemas.

“Cualquier modificación del software tiene una probabilidad no nula de introducir un nuevo fallo, y si esta probabilidad excede la reducción obtenida al fijar la infracción, el resultado neto es una mayor probabilidad de fallas en el software”(Boogerd & Moonen, 2009b).

En medio de todo esto se hacen dos preguntas:

¿Qué reglas usar para el estándar?

Un estándar puede ser simplemente ampliamente adoptado, pero todavía contienen reglas que, a simple vista no parecen las más adecuadas para un proyecto, además la simple existencia de diferentes estándares de codificación, se evalúa explícitamente las reglas individuales que pueden ayudar a valorar el estándar correcto.

¿Cómo priorizar una lista de violaciones?

Aunque un estándar puede ser cuidadosamente elaborado y personalizado para un proyecto, los desarrolladores se enfrentan a demasiadas violaciones demasiado fijas dado el tiempo limitado. Para manejar este problema de la manera más eficiente, se hace énfasis en una lista de violaciones, para definir un umbral para determinar cuáles deben ser abordados, y cuáles pueden omitirse.

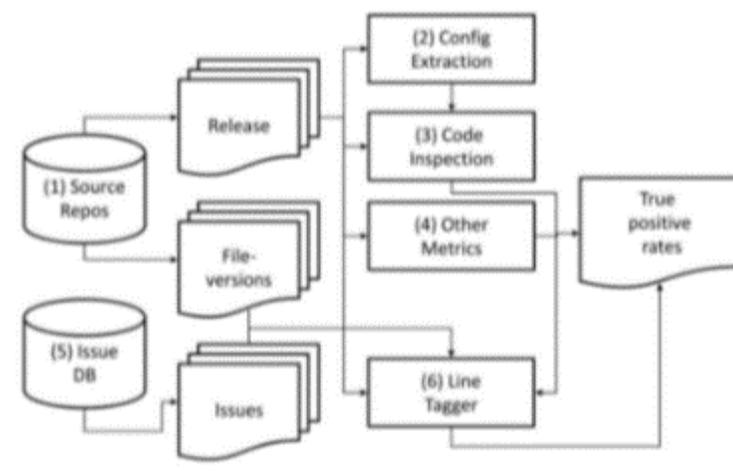


Figura 2. Visión general del proceso de medición(Boogerd & Moonen, 2009a).

Las mediciones de las infracciones de la figura 2 ilustra los pasos a seguir para el proceso de medición,

Primero se selecciona el rango de liberaciones para el estudio, es decir las partes del proyecto seleccionado, se hace una iteración sobre las recuperaciones del código fuente (1).

Se realiza una serie de operaciones para cada una, de una liberación se extrae la información de la configuración (2) necesaria para ejecutar la inspección automática de código (3) que a su vez mide el número de violaciones de código.

Se incluye un conjunto de archivos que deben formar parte de las mediciones para la construcción del proyecto.

Se graba el conjunto de archivos y se toman algunas medidas adicionales (4) que a su vez incluye el número de líneas de código de lanzamiento.

Recopila información con las versiones de archivo asociadas con errores presentes en la base de datos de los problemas (5).

Se procede a calcular una diferencia con las revisiones anteriores, revisando los cambios para resolver el problema y marcar las líneas modificadas.

se recorre finalmente para determinar de propagar las violaciones y cuáles estaban implicados en corregir, haciendo todo esto en la línea de tagger (6)(Boogerd & Moonen, 2008).

Finalmente los aspectos más representativos de un estándar de codificación hacen limitar al lenguaje a un conjunto seguro para evitar el uso de instrucciones inseguras, reduciendo eficazmente los defectos del código y mejorando la calidad de software, una de las normas de codificación ampliamente aceptadas para el software es MISRA-C.

MISRA-C es un conjunto de directrices para el uso de programación en este caso en C, primeramente, publicado en 1998 y realizando así una nueva versión en 2004, contiene 141 reglas de estándares de programación conveniente para el desarrollo crítico. Aborda directamente un diseño inadecuado para mejorar la capacidad del sistema, ofreciendo un servicio con fiable y logrando de esta manera ser aceptado por industrias y aplicaciones como lo son: militar, automóvil, aeroespacial, medico etc.

Para la comprobación de reglas relacionadas con las variables MISRA-C desensamblada el código fuente siendo implícito lo que hace que logre representar las variables de maneras operadas en código de desensamble, como resultado puede colocarse en un registro general o en una pila y referenciada por su dirección, siendo total mente diferente al código fuente. Una variable de código fuente hace desaparecer el código de desensamble debido a las optimizaciones del compilador, todo esto significa que algunas

reglas de estándar de codificación con las variables no pueden ser verificadas(Dai, Mao, Wang, Liu, & Zhang, 2010).

2.2 Línea de investigación

Se presenta modulo para la evaluación de estándares de codificación para la plataforma web de evaluación de productos de software de la universidad de Cundinamarca bajo la línea de investigación Software, Sistemas Emergentes y Nuevas tecnologías.

La línea de investigación busca el desarrollo de nuevo software, que implemente tecnologías modernas, usando estándares de calidad que permitan construir software moderno y de alta calidad.

2.3 Planteamiento del problema

En la actualidad la Universidad de Cundinamarca cuenta con la plataforma web para la evaluación de productos de software (CALISOFT). Con la cual se busca facilitar la evaluación de los parámetros de calidad de los productos de software de la universidad. Esto con el propósito de buscar una mayor calidad en los proyectos hechos por los estudiantes.

Cuando se habla de calidad son varios los factores que tienen que intervenir para que se pueda garantizar un producto de alta calidad. Haciendo referencia a la ISO/IEC 25010 está define un modelo de sistema y calidad del software basado en ocho características para realizar la evaluación: una de estas características es la mantenibilidad, el que un software pueda ser interpretado y modificado por alguien diferente a quien lo elaboro inicialmente facilita el poder cubrir esta característica establecida por el estándar, para lograr esto se recomienda aplicar estándares de programación en el producto que se está desarrollando.

El funcionamiento actual de la plataforma se basa en realizar las pruebas de tipo funcional sobre el aplicativo para verificar el correcto funcionamiento de este, pero esta no cuenta con una herramienta que permita evaluar que se aplique un estándar de programación predefinido, lo cual imposibilita poder evaluar todos los parámetros de calidad establecidos para el modelo de la ISO/IEC 25010.

¿Puede una herramienta informática evaluar la codificación de los productos de software desarrollados por el Centro de Innovación y Tecnología de la Universidad de Cundinamarca basados en un estándar de programación?

2.4 Objetivos

2.4.1 Objetivo general

Construir una herramienta que evalúe los estándares de codificación orientados a los productos de software generados por la universidad de Cundinamarca.

2.4.2 Objetivos específicos

- Hacer levantamiento de requerimientos del sistema.
- Construir el modelado y diseño del sistema propuesto.
- Determinar el estándar de codificación para los productos de software.
- Definir una propuesta de evaluación basado en el estándar seleccionado.
- Desarrollar el módulo para la evaluación de estándares de codificación.
- Realizar Pruebas el módulo anteriormente desarrollado, con diferentes muestras y diferentes tipos de escenarios de prueba.

2.5 Impacto del proyecto

Con el desarrollo de este proyecto se busca establecer la implementación de una herramienta que permita realizar la evaluación del estándar de programación predefinido para los proyectos hechos por los estudiantes de ingeniería de sistemas de la universidad de Cundinamarca, teniendo en cuenta lo importante que resulta este aspecto a la hora de desarrollar un software de calidad.

El poder contar con un instrumento que permita evaluar si se está aplicando el estándar de la forma adecuada, ayudara a los evaluadores de los proyectos el poder medir si está obteniendo un producto de alta calidad, acorde a lo que se espera de un desarrollo de esta envergadura como lo es un proyecto de grado.

Cuando se realizan desarrollos por etapas en el que un grupo de trabajo se encarga de la primera parte del desarrollo, el poder evaluar que este equipo haya utilizado el estándar de la forma correcta, permite que cuando el nuevo grupo continúe con la siguiente etapa, les sea más fácil el poder comprender lo que se ha hecho hasta el momento y poder continuar con la construcción del proyecto de la mejor forma.

2.6 Metodología

En la actualidad, en el ámbito científico se ha hecho necesario, dada la amplia gama de conocimientos desarrollados en cada una de las disciplinas, la aplicación de herramientas tecnológicas que ayuden a la organización, gestión y manejo de los datos para su posterior uso y aplicación en determinado contexto.

La metodología de investigación que se propone para la ejecución, realización y evaluación de estándares de codificación es la metodología mixta, refiriéndose a las dadas cualitativas y cuantitativas.

¿Por qué es cuantitativa?

El sistema recogerá información mediante cuestiones cerradas que se plantean de forma idéntica y homogénea, lo que permite su cuantificación y tratamiento estadístico por medio de pruebas funcionales de verificación hacia la calidad de software. Realiza una búsqueda referente a la generalización de los resultados a partir de muestras dentro de los márgenes de confianza y error (Universidad de Sonora, n.d.).

¿Por qué es cualitativa?

Debido a que la investigación de tipo cualitativa ha sido una de las grandes beneficiadas con la incursión de las tecnologías de la información aplicables a datos de análisis complejos y rigurosos como los cualitativos que difícilmente pueden ser categorizados y analizados como los trabajados en áreas del conocimiento como la antropología, sociología, trabajo social, psicología y las demás ramas de trabajo en las ciencias humanas.

En este contexto del enfoque cualitativo el sistema logrará organizar, analizar y encontrar perspectivas en datos no estructurados como: entrevistas, respuestas de encuestas con preguntas abiertas, artículos, contenido de las redes sociales y la web utilizando NVIVO (Pulido & Rodríguez, 2014).

2.7 Marcos de referencia

2.7.1 Marco Histórico

Cuando un nuevo proyecto empieza, o se utiliza un proyecto antiguo, el equipo de desarrollo se enfrenta a un nuevo paradigma, a menudo esto significa que se les pide que se adhieran a un nuevo conjunto de estándares de programación sujetándose a lo que se llama código por revisión por pares, siendo este proceso un enfoque práctico para la aplicación de normas y cuyo objetivo principal es la búsqueda de defectos y la propuesta de evaluación hacia las soluciones alternativas o de mejoras.

En un entorno de mantenimiento de código, un programador debe ser capaz de realizar cambios en un programa comprendiendo todo el proceso, ya que si opta por seguir las buenas prácticas de programación le será más fácil entender el código en un futuro. La mayoría de los estándares son reglas que hacen que el código sea más fácil de leer en la página, más fácil de seguir la lógica y ramas lógicas, y dejan menos espacio para la interpretación. El código que cumple estos requisitos es más fácil mantener. Los estándares son útiles trabajando en un ambiente de desarrollo de trabajo en equipo porque los requisitos mínimos, que a su vez aseguran cierta uniformidad de programador a programador. Son imperativo en la gestión de un gran proyecto, donde el control de código fuente también es un problema(Louise, 2009).

La investigación en el uso de estándares de programación no es algo recientemente, muchos autores y muchas compañías de desarrollo han optado por la creación de estos estándares sobre definiciones de estilos de programación medibles siendo muy activa en la década de 1980, cuando se crearon algunos modelos básicos y aplicaciones de evaluación bien conocidos.

Publicando Rees en 1982 la primera obra de estándares de programación que en el cual desarrollo un sistema STYLE para evaluar el estilo de programación en pascal sugiriendo que sean cuantificables y medibles por medio de 10 mediciones.

Basado en el trabajo de Rees, Berry y Meekings (1985) desarrollaron un analizador de estilo para comprobar el estilo de programación de C y probarlo con 80000 líneas de código de programa C. Su trabajo ha sido la base de muchas herramientas de evaluación de estilo de programa C, por ejemplo, Ceilidh (Benford, Burke, & Foxley, 1992) y modificado para Ada en ASSYST (Jackson & Usher, 1997). En estos sistemas, los valores de los parámetros para las mediciones son generalmente configurables por los profesores del curso. En Ceilidh, el modelo se refina más para incluir también comentarios de texto para los estudiantes, es decir, dar cierto feedback si el valor medido está por debajo del hitol o sobre el hitol(Michaelson, 2015).

Más adelante, Redish y Smyth (1986) desarrollaron un analizador del estilo de Fortran para ayudar con clases grandes. Se acercaron al problema clasificando las mediciones bajo cualidades estilísticas como economía, modularidad, simplicidad, estructura, documentación y diseño. Su programa de análisis contenía un total de 33 mediciones, incluyendo tamaños de bloque, número de operadores, número de declaraciones y medidas de complejidad. Otros programas con fines educativos también se han implementado, por ejemplo, CAP (Schorsch, 1995) para la autoevaluación de la sintaxis, la lógica y el estilo en los programas Pascal y la herramienta de análisis de Michaelson (1996) para programas funcionales.

Omán y Cook (1990) recopilaron información de diferentes pautas de estilo y analizadores. Propusieron una taxonomía de estilo para categorizar factores estilísticos en las prácticas de programación General, Estilo tipográfico, Estilo de estructura de control y Estilo de estructura de información. Sugieren que en lugar de un conjunto de reglas de estilo separadas, los estudiantes deben ser enseñados principios de estilo de programación, según su taxonomía. Su trabajo ha sido referido a menudo en enfoques educativos posteriores.

En la investigación de ingeniería de software, Dromey (1995) publicó su modelo sobre la conexión de atributos de calidad del software a las propiedades del programa. PASS-C (SQI, 1993) puede analizar 99 mediciones de C dando retroalimentación que conecta los defectos medidos con los atributos de la calidad del software. Analizadores similares también se implementan para los lenguajes Ada y Java. Es posible ver en su trabajo las conexiones entre la calidad general del software y las diferentes reglas de estilo. Aunque el enfoque es muy diferente, también se pueden ver algunas similitudes con las categorías de Omán y Cook (Ala-Mutka et al., 2004).

Actualmente se rigen los siguientes estándares de programación que sirven de apoyo en los diferentes proyectos de investigación siendo de gran ayuda para los equipos de trabajo de las diferentes empresas, por ejemplo.

Java Coding Conventions estándar de programación de java el cual brinda una serie de recomendaciones para tener buenas prácticas de codificación.

¿Por qué tener convenciones de código?

- Las convenciones de código son importantes para los programadores por una serie de razones:
- El 80% del costo de la vida útil de una pieza de software se destina al mantenimiento.
- Casi ningún software se mantiene durante toda su vida por el autor original.
- Las convenciones de código mejoran la legibilidad del software, permitiendo a los entender el nuevo código de forma más rápida y completa.
- Si envía su código fuente como un producto, debe asegurarse de que esté bien empaquetado y limpiar como cualquier otro producto que usted crea ("Java Coding Conventions," 1997).

El siguiente estándar de programación es el de C# estándares de codificación para .NET, este documento describe las reglas y recomendaciones para el desarrollo de aplicaciones y bibliotecas de clases que utilizan el lenguaje C#. El objetivo es definir directrices para reforzar el estilo y el formato coherentes y ayudar a los desarrolladores a evitar trampas y errores (Hunt, 2012).

Continuando con los supuestos, otro estándar de programación que ya se había mencionado es el MISRA, siendo compilaciones de directrices para la codificación en los lenguajes C y C ++. Se utiliza ampliamente en el desarrollo de sistemas críticos de software cuando se deben cumplir los requisitos de un estándar de calidad. Muchos proyectos de software especifican que la calidad del código debe asegurarse cumpliendo con los requisitos de las Directrices. Sin embargo, el significado de la frase "compatible con MISRA" necesita ser cuidadosamente definido.

Para que un reclamo de cumplimiento MISRA tenga sentido, establecen:

- Exactamente qué directrices se están aplicando.
- La eficacia de los métodos de aplicación.
- a medida en que se han utilizado las desviaciones.
- Uso de un proceso disciplinado de desarrollo de software.
- El estado de cualquier componente desarrollado fuera del proyecto(MISRA, 2016).

Al igual que la evolución que han tenido los lenguajes de programación, los estándares desarrollados para estos también han tenido un crecimiento a lo largo de la historia y son altamente utilizados en la actualidad, su uso se ha masificado y son de gran valor para las compañías que se dedican al desarrollo de software, los beneficios que estos brindan son ampliamente conocidos y su desarrollo se seguirá extendiendo, es por esto que es importante conocerlos y aplicarlos de una forma adecuada para obtener un código limpio y que sea fácilmente mantenible para así lograr un software de calidad.

2.7.2 Marco teórico

Actualmente en la Universidad de Cundinamarca existe una herramienta de trabajo para el proceso de evaluación de la calidad de software llamado Calisoft, el cual se fundamenta en una plataforma de evaluación para los proyectos de grado que se basan en la realización de software mediante tres (3) sistemas de calificaciones. El primer sistema de calificación es en donde se presenta al evaluador la documentación y modelación del proyecto, el segundo sistema es la parte en donde se evalúa la gestión de pruebas tanto funcionales como prueba de carga y estrés, la tercera herramienta cuenta con la parte administrativa en donde se hacen las configuraciones de acuerdo con los estándares de calidad.

El módulo que se propone es un complemento al sistema de evaluación de gestión de pruebas, lo cual mejorara el sistema de calificación de la plataforma de Calisoft, este complemento dará un gran soporte ante los estándares de

programación que se rigen actualmente en el mundo, se busca que con la implementación de este módulo los evaluadores puedan garantizar que se esté respetando el estándar de codificación previamente establecido, mientras que el módulo de proyecto de estandarización determine una calificación hacia el desarrollador, de esta manera se puede constatar en que posibles errores o en qué aspectos el estudiante está fallando a la hora de codificar, recordando la importancia que tiene este ítem para garantizar un software de calidad.

2.7.2.1 Patrón de diseño MVC

El patrón de diseño MVC fue creado con el objetivo de disminuir el esfuerzo de programación en el desarrollo de sistemas a gran escala, una de las principales características del patrón, es que tanto las vistas, los modelos y los controladores se tratan como entidades separadas, esto permite que cualquier cambio que se haga en cualquier de los componentes no afecte a los demás, permitiendo así que cada uno de estos pueda ser desarrollado por separado, esto elimina el problema de que si algún componente no funciona de la manera correcta pueda ser cambiado, sin la necesidad de hacer cambios en los otros elementos del sistema.

Los modelos en el patrón MVC son los que representan los datos del sistema, estos se encargan del manejo de los datos y de cualquier transformación que estos sufran, el modelo es independiente de los controladores y las vistas, este no contiene referencias a ellos y es responsabilidad del sistema crear los enlaces para comunicarlos entre sí.

La vista es el objeto que se encarga de la presentación visual de los datos representados por el modelo. Este genera una representación de los datos y se encarga de presentarlos al usuario. Interactúa la mayor parte del tiempo con el controlador, aunque en ocasiones puede relacionarse directamente con el modelo.

El controlador es el objeto que atiende a las órdenes dadas por el usuario, actúa sobre los datos del modelo, realiza la interacción entre la vista y el modelo. Se activa cuando existe un cambio en la vista y mantiene una comunicación con el modelo a través de una referencia directa a este (Fernández Romero Yenisleidy, 2012).

2.7.2.2 Estándar de codificación

Los estándares de codificación son una serie de convenciones para escribir código fuente en un determinado lenguaje de programación. A través de un estándar se pueden definir reglas tales como, el tipo de nomenclatura o

notación que se debe usar para nombrar los elementos dentro del código como variables, constantes, clases etc. Además especifica como estructurar el código, cuando usar una determinada estructura y demás reglas que se necesiten de acuerdo al objetivo del proyecto a desarrollar.

El uso de estándares de codificación en el desarrollo de software permite tener un código limpio y ordenado, que sea fácil de leer y mantener por cualquier miembro del equipo de desarrollo, buscando tener no solo un software que trabaje si no un software que trabaje bien.

2.7.3 Marco legal

Los desarrollos de los productos de software deben estar soportados por normas de calidad que hayan sido emitidas por organizaciones internacionales como la ISO teniendo en cuenta que la evaluación generada cumpla con las normas internacionales como la ISO/IEEE 25010, buscando establecer las pautas mínimas que debe cumplir un software de calidad.

La norma ISO/IEEE 25010 establece en modelo de calidad de software, un sistema de calificación de los productos, determinando las características de calidad que se tienen en cuenta a la hora de evaluar las propiedades de un producto de software determinado.

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad.

- Adecuación función
- Eficiencia de desempeño
- Compatibilidad
- Usabilidad
- Fiabilidad
- Seguridad
- Mantenibilidad
- Portabilidad

El cual los productos de software están ligados por la parte de la mantenibilidad el cual representa su modificación efectiva y eficiente debido a las necesidades evolutivas o perspectivas.

Para el desarrollo del producto de software se implementó herramientas de software libre las cuáles están regidas por las siguientes licencias:

2.7.3.1 Laravel

Es un framework de código abierto para desarrollar aplicaciones web bajo el lenguaje de programación PHP, permite el uso de sintaxis elegante, sencilla y con múltiples funcionalidades aprovechando lo mejor de otros frameworks como Symfony y las últimas versiones de PHP. Laravel cuenta con la licencia del MIT (Massachusetts Institute of Technology) y llamada también licencia X11, ya que es la licencia que llevaba este software de muestra de la información de manera gráfica X Windows System originario del MIT en los años 1980. Pero ya sea como MIT o X11, su texto es idéntico.

El texto de la licencia no tiene copyright, lo que permite su modificación. No obstante esto, puede no ser recomendable e incluso muchas veces dentro del movimiento del software de código abierto desaconsejan el uso de este texto para una licencia, a no ser que se indique que es una modificación, y no la versión original.

2.7.3.2 MySQL

Es un sistema de gestión de base de datos desarrollado bajo la licencia de código abierto más popular del mundo para entornos de desarrollos web. Existen diferentes interfaces de programación que permiten el acceso a la base de datos a aplicaciones escritas en lenguajes de programación que soporte ODBC como PHP. MySQL cuenta con la licencia GNU/GPL, permitiendo la libertad de usar, estudiar, compartir (copiar) y modificar el software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. Esta licencia fue creada originalmente por Richard Stallman fundador de la Free Software Foundation (FSF) para el proyecto GNU.

2.7.3.3 Vue js

Es un framework de JavaScript progresivo de código abierto para crear interfaces de usuario. fue creado para organizar y simplificar el desarrollo web, contiene una arquitectura modelo-vista-controlador, sirve principalmente como la vista, mientras que también aborda algunas preocupaciones manejadas convencionalmente por el controlador permitiendo a los usuarios recibir actualizaciones de datos en vivo e interactuar con los datos a través de métodos específicos de componentes. De la misma manera que MySQL, Vue.js cuenta con la licencia GNU/GPL.

3.2 Determinación de requerimientos

3.2.1 Requerimientos funcionales

Número de requisito	RF 01
Nombre de requisito	Generar los parámetros de estándares
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Esta actividad permite definir como serán los parámetros de estándares para la codificación, para luego posterior mente evaluar las declaraciones generales.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 02
Nombre de requisito	Evaluar declaración de variables
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Esta actividad permite realizar el recorrido de las variables que han sido declaradas en el archivo fuente que se está evaluando, para su calificación y posterior aprobación frente al evaluador
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 03
Nombre de requisito	Evaluar declaración de clases
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Esta actividad permite realizar el recorrido de las clases que han sido declaradas en el archivo fuente que se está evaluando, para su calificación y posterior aprobación frente al evaluador
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 04
Nombre de requisito	Evaluar declaración de funciones
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Esta actividad permite realizar el recorrido de las funciones que han sido declaradas en el archivo fuente que se está evaluando, para su calificación y posterior aprobación frente al evaluador
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 05
Nombre de requisito	Evaluar declaración de constantes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Esta actividad permite el recorrido de la sangría que se esté utilizando por el estándar, con respecto a su declaración en el archivo fuente que se está evaluando, para su calificación y posterior aprobación frente al evaluador
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 06
Nombre de requisito	Evaluar declaración de indentación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Esta actividad permite el recorrido de la sangría que se esté utilizando por el estándar, con respecto a su declaración en el archivo fuente que se está evaluando, para su calificación y posterior aprobación frente al evaluador
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 07
Nombre de requisito	Evaluar declaración de comentarios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Esta actividad permite realizar la evaluación de comentarios en el archivo fuente, para su calificación y posterior aprobación frente al evaluador
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 08
Nombre de requisito	Evaluar declaración de espacios de nombre
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Esta actividad permite realizar el recorrido de los espacios de nombre que han sido declaradas en el archivo fuente que se está evaluando, para su calificación y posterior aprobación frente al evaluador
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 09
Nombre de requisito	Crear Rubrica de evaluación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Esta actividad permite asignar un valor o un porcentaje de evaluación a cada una de las evaluaciones estipuladas en el módulo.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 10
Nombre de requisito	Sistema de calificaciones
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Esta actividad permite que el sistema genere una calificación basado en la rúbrica de evaluación.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 11
Nombre de requisito	Creación de la interfaz Gráfica
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Está actividad permite la creación y el funcionamiento respectivo de la interfaz gráfica para el módulo, de acuerdo al estándar establecido por el centro de innovación y tecnología (CIT)
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 12
Nombre de requisito	Visualización del código fuente
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Está actividad permite generar una visualización del código fuente del producto evaluado con la finalidad de tener una interfaz amigable por parte del evaluador, para el manejo de los módulos de calificación de estándares de programación
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 13
Nombre de requisito	Campo de comentarios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Está actividad permite al evaluador tener un complemento para dejar sus opiniones, aportes y observaciones de los productos de software que se evalúan en el modulo
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 14
Nombre de requisito	Informes de calificación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Está actividad permite que el modulo genere informes de calificación con las diferentes pruebas de evaluación que se especifican en el modulo También cuenta con la visualización de las observaciones por parte del evaluador Y por último contara con la opción de descarga de PDF
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 15
Nombre de requisito	Reporte de modulo general
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Está actividad permite generar un informe estadístico de los diferentes proyectos que se han evaluado hasta la fecha, con el fin de visualizar los errores generales presentados en los productos que se han evaluado
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF 16
Nombre de requisito	Integración del módulo con la plataforma Calisoft
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Esta actividad, permite integrar el módulo, con la plataforma Calisoft, de tal forma que, al evaluar un proyecto, se dará un porcentaje correspondiente, pero teniendo en cuenta Calisoft y los diferentes módulos.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

3.2.2 Definición de requerimientos funcionales

3.2.2.1 Requerimiento funcional 1

Tabla de categorías: para el administrador encargado, contara con una tabla de categorías lo cual tiene los módulos de porcentajes referentes a plataforma, modelado, bases de datos y codificación, del mismo modo cada categoría que el administrador desee agregar a la plataforma, la suma total de estos cuatro (4) módulos será 100% y podrá agregar modificar eliminar cada uno de las categorías para la calificación del módulo de codificación.

3.2.2.2 Requerimiento funcional 2

Tabla importancia de ítems: para el administrador encargado de la plataforma de Calisoft contara con una tabla de ítems, entre estos variables, clases, funciones, constantes, indentación comentarios y espacios de nombre, lo cual cuentan con un valor. Este valor es una matriz de grado de importancia para las calificaciones individuales del estudiante en cuanto a los ítems y que sirve como apoyo para la nota de estudiante.

3.2.2.3 Requerimiento funcional 3

Carga de archivos: el estudiante que por el cual el administrador le haya aprobado el proyecto, tendrá una carga de archivos y que podrá subir los scripts o archivos (. PHP) teniendo cada script un límite de 10 MB.

3.2.2.4 Requerimiento funcional 4

Tabla de la carga de archivos: luego de que el estudiante haya cargado todos sus archivos, contara con una tabla para la revisión de sus archivos, para no tener un fallo, si tiene un fallo de archivo tendrá la opción de eliminar el archivo que no pertenece, está tabla tiene como columnas el nombre, el estado del documento y las operaciones referentes a este mismo.

3.2.2.5 Requerimiento funcional 5

Proyectos asignados: luego de que el administrador haya asignado un proyecto a un docente, el docente contara con un panel de proyectos asignados y que por el cual aparecerán cuatro (4) botones los cuáles son los módulos correspondientes de plataforma, modelación, codificación y base de datos.

3.2.2.6 Requerimiento funcional 6

Panel de vista codificación: luego que el docente se dirija a la parte de codificación, se tendrá una vista del proyecto del mismo modo tendrá un panel de archivos (. PHP) que el estudiante subió de su proyecto para su posterior calificación.

3.2.2.7 Requerimiento funcional 7

Revisión del estándar de codificación: el docente tendrá una sección de lectura que se encontrará antes de calificar cada proyecto en caso que no haya revisado el estándar de codificación para su posterior calificación.

3.2.2.8 Requerimiento funcional 8

Revisión de tablas de porcentajes: tanto el estudiante como el docente tendrán un panel para la revisión de porcentajes generalizada por el administrador y un panel de porcentajes por modulo para la respectiva calificación.

3.2.2.9 Requerimiento funcional 9

Ejecución y vista de la calificación: el docente tendrá un panel de revisión del script, lo cual podrá observar el código de proyecto del estudiante, luego ejecutara un programa de revisión de código para la calificación del estudiante que está validado por la matriz de importancia para sacar su posterior nota de cada uno de los ítems de la tabla; si el docente está de acuerdo acepta la nota o si la rechaza tendrá una caja de texto lo cual podrá modificar la nota.

3.2.2.10 Requerimiento funcional 10

Caja de comentarios: cuando el docente haya determinado la nota del estudiante tendrá una caja de texto opcional para dejar sus comentarios y aportaciones al estudiante.

3.2.2.11 Requerimiento funcional 11

Envío de la nota al estudiante: cuando si el docente deja o no comentario cuando ya haya determinado la nota enviara sus aportes al estudiante lo cual le notificara al estudiante cuando ya lo haya calificado y en la tabla de scripts del estudiante cambiara el estado de enviado ha calificado.

3.2.2.12 Requerimiento funcional 12

Descarga de PDF docente: el docente encargado podrá descargar el PDF con los reportes del módulo de codificación por proyecto o por registro extendido global para determinar las falencias de los proyectos para la recomendación de cada proyecto.

3.2.2.13 Requerimiento funcional 13

Descarga de PDF estudiante: el estudiante podrá descargar el reporte en PDF del módulo de codificación en PDF para determinar sus notas, sus falencias y aportes del docente para su mejora.

3.2.2.14 Requerimiento funcional 14

Descargar General PDF estudiante: el estudiante podrá ver la nota general de calificación de los cuatro (4) módulos de Calisoft para mirar su nota final y los aportes del profesor.

3.2.2.15 Requerimiento funcional 15

Paneles de ayuda: En cada una de las vistas del módulo de codificación de Calisoft contara con un link de ayuda lo cual abre un modal para la lectura y manejo de la interfaz.

3.2.3 Requerimientos no funcionales

3.2.3.1 Requerimientos de rendimiento

Para determinar los requisitos de rendimiento primero se toma el rendimiento del servidor que soportara la plataforma de Calisoft que por la cual aún no se han especificado, lo segundo se harán pruebas de carga y estrés de la

plataforma para determinar las falencias y mejora de la plataforma que por la cual tampoco no se han determinado.

3.2.3.2 Seguridad

La plataforma de Calisoft está creada bajo el framework laravel y lo cual cuenta con los siguientes requerimientos de seguridad.

- Intento de phishing.
- Storing Passwords - laravel proporciona una clase llamada "Hash" de clase que proporciona hash Bcrypt seguro.
- CSRF Protection/Cross-site request forgery (XSS).
- vulnerabilidad de inyección SQL existe cuando una aplicación inserta la entrada del usuario arbitraria y no filtrada en una consulta SQL.
- Cookies – Secure by default - laravel hace que sea muy fácil de crear, leer, y expiran las galletas con su clase de galletas.
- HTTPS evita que los atacantes en la misma red para interceptar información privada, como variables de sesión, y entre como la víctima.

Además, la plataforma cuenta con todas las validaciones posibles para evitar la caída del sistema.

3.2.3.3 Fiabilidad

El sistema de seguridad de la plataforma, sus componentes y sus validaciones son útiles para mejorar la fiabilidad del sistema. Laravel soporta diferentes modos de registro como única, diaria, syslog, y los modos del registro de errores para el rendimiento del sistema.

3.2.3.4 Disponibilidad

La plataforma de Calisoft contara con un servidor 100% disponible para el acceso a las 24 horas.

3.2.3.5 Mantenibilidad

Para el mantenimiento de la plataforma se necesita un programador con el manejo del framework de laravel, con conocimientos en VUE.js, JavaScript, y la base de datos en MYSQL.

Se especifica el mantenimiento cada vez que se actualiza el framework de laravel para que la plataforma adquiera todos los requerimientos disponibles por actualización.

3.2.3.6 Portabilidad

La plataforma cuenta con Responsive, lo cual le permite a la persona total acceso al sistema desde los diferentes dispositivos que cuenten con internet para el manejo de la plataforma, también el proyecto de Calisoft cuenta con su sistema de versionamiento en GITHUB, para el manejo de repositorios y clonación de la plataforma.

3.2.4 Requisitos funcionales de las interfaces

3.2.4.1 Interfaces de usuario

La interfaz del usuario consistirá en un conjunto de ventanas con botones y menús desplegables, será construida específicamente para el módulo propuesto y será en entorno web, es decir, que podrá ser visualizada en un navegador de internet.

3.2.4.2 Interfaces de hardware

- Monitor
- Mouse
- Teclado
- Memoria mínima de 256Mb
- Procesador de 1.66GHz o superior
- Adaptador de red

3.2.4.3 Interfaces de software

- Sistema Operativo: Multiplataforma.
- Explorador: Mozilla, Chrome, Opera, Safari, Internet Explorer.

3.2.4.4 Interfaces de comunicación

Los interfaces de comunicación serán, los servidores, clientes y aplicaciones que se comuniquen entre sí, mediante protocolos en internet (FTP).

4. Especificación del sistema

Para la realización y ejecución del módulo para la evaluación de estándares de codificación se tuvieron en cuenta 2 investigaciones importantes.

4.1 Estándar de codificación

La Investigación que se presta a continuación es un manual de uso del estándar de codificación seleccionado por el semillero de “aplicaciones móviles” para el desarrollo de aplicativos webs en la universidad de Cundinamarca, que utilicen el lenguaje de programación de PHP.

Se establecen 7 directrices de calificación que son los más relevantes en el momento de escribir código entre las cuáles se presentan a continuación.

4.1.1 Variables

La notación recomendada para la declaración de variables es la notación lowerCamelCase está consiste en que cuando las variables formadas por más de una palabra cada una de estas deberá iniciar en mayúscula exceptuando la primera palabra, si la palabra conlleva guiones bajos en una posición diferente a la inicial la declaración será errónea, también la longitud mínima de una variable es de 3 caracteres exceptuando las variables \$i, \$j, \$k que se utilizan en los ciclos “for”.

```
1  /**
2   * ejemplo de declaracion de variables utilizando la notacion lowerCamelCase
3   */
4
5  <?php
6  $nombreEstudiante = "Malcolm Young";
7  $saldoEmpleado = 700000;
8  $porcentajeDeAuxilio = 0.13;
9
```

Figura 4. Ejemplo declaración de variables.

4.1.2 Clases

La notación recomendada para la declaración de clases es la notación UpperCamelCase está consiste en que cuando las clases formadas por más de una palabra cada una de estas deberá iniciar en mayúscula, si la clase tiene una sola palabra su primera letra estará en mayúscula, si la palabra conlleva guiones bajos en una posición diferente a la inicial la declaración será errónea, también la longitud mínima de una clase es de 3 caracteres.

```

1  /**
2   * ejemplo de declaracion de clases utilizando la notacion UpperCamelCase
3   */
4
5  <?php
6
7  class Persona{
8     //aqui va el codigo
9  }
10
11 class ClaseDeProgramacion{
12     //aqui va el codigo
13 }
14

```

Figura 5. Ejemplo declaración de clases.

4.1.3 Funciones

La notación recomendada para la declaración de funciones es la notación lowerCamelCase está consiste en que cuando las funciones formadas por más de una palabra cada una de estas deberá iniciar en mayúscula exceptuando la primera palabra, si la palabra conlleva guiones bajos en una posición diferente a la inicial la declaración será errónea, también la longitud mínima de una función es de 3 caracteres.

```

1  /**
2   * ejemplo de declaracion de funciones utilizando la notacion lowerCamelCase
3   */
4
5  <?php
6
7  function mostrarFunciones(){
8     //aqui va el codigo
9  }
10
11 function traerAlgunasVariables($variable){
12     //aqui va el codigo
13 }
14

```

Figura 6. Ejemplo declaración de funciones.

4.1.4 Constantes

La notación recomendada para la declaración de constantes es la notación Snake Case está consiste en que las constantes serán declaradas totalmente en mayúsculas y si esta está compuesta por más de una palabra debe estar

separada por un underscore o guion abajo, también la longitud mínima de una constante es de 3 caracteres.

```
1  /**
2   * ejemplo de declaracion de constantes utilizando la notacion Snake Case
3   */
4
5  <?php
6
7  class Clase{
8
9      const VALOR_MIN = 1.0; //constante de clase
10
11  }
12  define('VALOR_MIN','1.0'); //constante
13
```

Figura 7. Ejemplo declaración de constantes.

4.1.5 Indentacion

La indentación se puede expresar como la sangría que debe tener cada estructura de código.

Entiéndase por estructura de control “if”, “else”, “while”, “for”, etc. para este estándar se definió una indentación mínima de 4 espacios, lo que corresponde a una tabulación, aunque el espacio puede ser mayor, cabe resaltar que cada elemento que este dentro de la estructura debe respetar el mismo espaciado que los demás, toda estructura declarada debe siempre llevar las llaves de apertura y cierre aun cuando esté compuesta por una sola sentencia y la sentencia debe ir después de la estructura (no pueden ir en la misma línea).

```

1
2  /**
3  * ejemplo de uso de la indentacion de estructuras
4  */
5  <?php
6  $variableA=5;
7  $variableB=10;
8      //forma correcta de indentacion
9      if($variableA < $variableB){
10         echo $variableA;
11         $variableA++;
12     }else{
13         echo $variableB;
14         $variableB++;
15     }

```

Figura 8. Forma correcta de indentacion.

```

16      //forma incorrecta de indentacion
17      for($i=0;$i<$variableA+$variableB;$i++){
18  echo "la iteracion de la variable i es: ".$i;
19         $variableA=$i+$variableB;
20         echo $variableA+$variableB;
21     }
22
23  ?>

```

Figura 9. Forma incorrecta de indentacion.

4.1.6 Comentarios

Cuando se declare una clase o una función debe ser comentada en su cabecera, el no realizar un comentario la declaración será errónea.

```

1
2  /**
3   * ejemplo del uso de comentarios en una clase y una funcion
4   */
5  <?php
6
7   //esta clase me permite realizar una suma
8   class RealizacionSuma{
9       //esta funcion me retorna una suma
10      function Suma(){
11          $variableA=10;
12          $variableB=20;
13          return $variableA+$variableB;
14      }
15  }
16  ?>

```

Figura 10. Uso de comentarios en el código fuente.

4.1.7 Espacios de nombre

La notación recomendada para la declaración de espacios de nombre es la notación UpperCamelCase está consiste en que cuando los espacios de nombre formados por más de una palabra cada una de estas deberá iniciar en mayúscula, si el espacio de nombre tiene una sola palabra su primera letra estará en mayúscula, si la palabra conlleva guiones bajos en una posición diferente a la inicial la declaración será errónea, también la longitud mínima de un espacio de nombre es de 3 caracteres.

```

1
2  /**
3   * ejemplo de declaracion de espacio de nombre
4   */
5  <?php
6   namespace Persona;
7
8   class Persona{
9       //aqui va el codigo
10  }
11  ?>

```

Figura 11. Ejemplo declaración de espacios de nombre.

4.2 Métrica para la evaluación de estándares de codificación

Es importante analizar cada una de las directrices que propone el estándar explicado como lo es el PSR. Este estándar propone una serie de normas que aplicarlas al momento de codificar software permite que este sea más limpio y pueda ser fácilmente entendido por otro programador; Analizando las directrices que expone el estándar, se busca identificar cuáles de estas son de mayor utilidad y tienen mayor peso para lograr el objetivo de obtener un código de calidad y que pueda ser fácilmente mantenido.

Partiendo de los supuestos anteriores después de identificar que directrices son de mayor importancia, se busca elaborar la métrica de evaluación teniendo como categorías o elementos de evaluación cada una de las normas del estándar especificado y asignándole un porcentaje de calificación de acuerdo a la importancia o peso considerado; posteriormente se definirá los niveles de calificación para cada una de las categorías, estos deben ser cuantificables y medibles, con esto se busca generar una calificación la cual pueda estar basada en normas y estándares usados en la industria de software actual.

Con el diseño de esta métrica se busca poder evaluar al programador para verificar si está aplicando el estándar de la forma correcta, además el poder ser calificado permite al programador poder identificar en qué aspectos está presentando falencias al momento de codificar, con esto se busca lograr que el software entregado sea mucho mejor elaborado.

Para la construcción de la métrica de evaluación las categorías que se van a evaluar son algunas de las directrices propuestas por el estándar, las cuales son:

- Declaración de variables.
- Declaración de métodos.
- Declaración de clases.
- Indentación de las estructuras de control.
- Uso de comentarios en el código fuente.
- Declaración de constantes.
- Uso de espacios de nombre.

A cada una de estas directrices se le asigna una prioridad de acuerdo a su grado de relevancia dentro del estándar propuesto, con esto se busca que las directrices que son usadas con mayor frecuencia y que pueden ser de mayor importancia para lograr el correcto uso del estándar tengan un mayor porcentaje al momento de obtener la calificación final.

Directrices	Prioridad		
	Alta	Media	Baja
Variables	X		
Métodos	X		
Clases	X		
Indentación		X	
Comentarios		X	
Constantes			X
Espacios de nombre			X

Tabla 1. Prioridad de cada directriz.

Observando la tabla anterior se pueden identificar tres niveles de prioridades alta, media y baja. Cada una de estas tendrá un valor numérico que estará en el rango de 1 a 5 siendo las directrices de prioridad alta las que posean un mayor valor.

4.2.1 Ecuaciones

La forma de obtener el índice de cohesión final se basará en obtener la calificación individual de cada una de las normas, para luego consolidarlas en una única ecuación y obtener la nota correspondiente, para hacer este cálculo se utiliza una ecuación para cada una de las directrices anteriormente mencionadas, las cuales se presentan a continuación:

$$Nv = \frac{\sum_{N=0}^{NTA} \frac{NVC}{NVT}}{NTA} \quad (1)$$

Ecuación 1. Calificación de variables.

NTA: Número total de archivos de código de fuente donde se hayan declarado variables.

NVC: Numero de variables correctamente declaradas.

NVT: Numero de variables totales declaradas por archivo de código fuente.

Nv: Calificación para la directriz declaración de variables.

$$Nm = \frac{\sum_{N=0}^{NTA} \frac{NMC}{NMT}}{NTA} \quad (1)$$

Ecuación 2. Calificación de métodos.

NTA: Número total de archivos de código de fuente donde se hayan declarado métodos.

NMC: Numero de métodos correctamente declarados.

NMT: Numero de métodos totales declarados por archivo de código fuente.

Nm: Calificación para la directriz declaración de métodos.

$$Nc = \frac{\sum_{N=0}^{NTA} \frac{NCC}{NCT}}{NTA} \quad (2)$$

Ecuación 3. Calificación de clases.

NTA: Número total de archivos de código de fuente donde se hayan declarado clases.

NCC: Numero de clases correctamente declarados.

NCT: Numero de clases totales declaradas por archivo de código fuente.

Nv: Calificación para la directriz declaración de clases.

$$Ni = \frac{\sum_{N=0}^{NTA} \frac{NIC}{NIT}}{NTA} \quad (3)$$

Ecuación 4. Calificación de indentacion.

NTA: Número total de archivos de código de fuente.

NIC: Numero de estructuras de control correctamente indentadas.

NIT: Numero de estructuras de control utilizadas por archivo de código fuente.

Ni: Calificación para la directriz de indentación de estructuras de control.

$$N_{cm} = \frac{\sum_{N=0}^{NTA} \frac{CMC}{CMT}}{NTA} \quad (4)$$

Ecuación 5. Calificación de comentarios.

NTA: Número total de archivos de código de fuente.

CMC: Numero de estructuras de código comentadas.

CMT: Numero de estructuras utilizadas por archivo de código fuente.

Ncm: Calificación para la directriz uso de comentarios en el código fuente.

$$N_{cs} = \frac{\sum_{N=0}^{NTA} \frac{CSC}{CST}}{NTA} \quad (5)$$

Ecuación 6. Calificación de constantes.

NTA: Número total de archivos de código de fuente donde se hayan declarado constantes.

CSC: Numero de constantes correctamente declarados.

CST: Numero de constantes totales declaradas por archivo de código fuente.

Ncs: Calificación para la directriz declaración de constantes.

$$N_e = \frac{\sum_{N=0}^{NTA} \frac{NSC}{NST}}{NTA} \quad (6)$$

Ecuación 7. Calificación de espacios de nombre.

NTA: Número total de archivos de código de fuente donde se hayan declarado espacios de nombre.

NSC: Numero de espacios de nombre correctamente declarados.

NST: Numero de espacios de nombre declarados por archivo de código fuente.

Ne: Calificación para la directriz de uso de espacios de nombre.

El cálculo del índice final se realizará aplicando una única ecuación en la cual se consolidarán cada una de las calificaciones obtenidas para cada directriz y estas tendrán un peso dentro de la ecuación de acuerdo con su a nivel de prioridad.

$$Ic = \frac{Vp(Nv) + Vp(Nm) + Vp(Nc) + Vp(Ni) + Vp(Ncm) + Vp(Ncs) + Vp(Ne)}{Vp(Npa) + Vp(Npm) + Vp(Npb)} \quad (7)$$

Ecuación 8. Cálculo del índice de cohesión.

Vp: Valor de la prioridad de cada directriz.

Npa: Número de directrices de prioridad alta.

Npm: Número de directrices de prioridad media.

Npb: Número de directrices de prioridad baja.

Ic: Índice de cohesión.

La ecuación (8) se utiliza para poder calcular el índice de cohesión, para esto se necesita el valor obtenido para cada una de las directrices anteriormente explicadas, por ello cada uno de estos valores se multiplica por su respectiva prioridad y se sumaran, para luego ser divididos por el total de las directrices utilizadas en la codificación. El índice de cohesión busca medir a que nivel se está utilizando el estándar al momento de codificar, este valor puede ser representado como un porcentaje y así poder evaluar al programador sobre la aplicación de esta norma de calidad.

5. Especificación de diseño

Luego del cumplimiento de 2 objetivos importantes impuestos en el proyecto de grado los cuáles son la creación de una métrica de evaluación de estándares de codificación y un modelo de estándar de codificación, el siguiente paso para la creación del producto de software que evalué la correcta forma de programación de los estudiantes de la universidad de Cundinamarca es el modelado, el cual se despliegan.

- Modelo entidad relación.
- Diagramas de caso de uso.
- Diagramas de secuencia.
- Diagramas de actividades.
- Diagrama de clase.

5.1 Modelo entidad relación

EL modelo entidad relación es una abstracción del diseño de la base de datos, el cual está estructurado por tablas, relaciones, llaves primarias y llaves foráneas, este busca dar una estructura ordenada que permita evitar la redundancia de datos y agilizar la búsqueda de información al ser un modelo organizado.

La siguiente figura que se presenta a continuación es una abstracción de la forma en general como se presenta la base de datos de Calisoft.

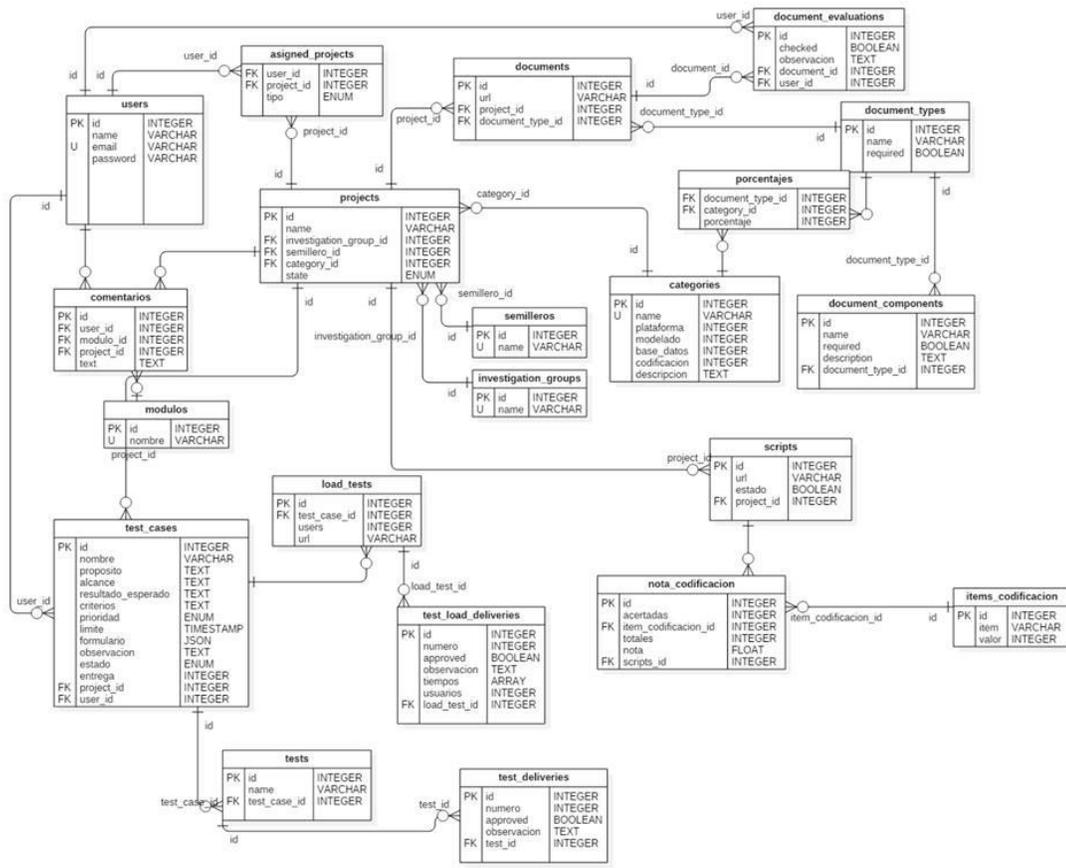


Figura 12. Modelo entidad relación general.

La siguiente figura que se presenta del modelo entidad/relación es la segmentación solo para la evaluación de estándares de codificación:

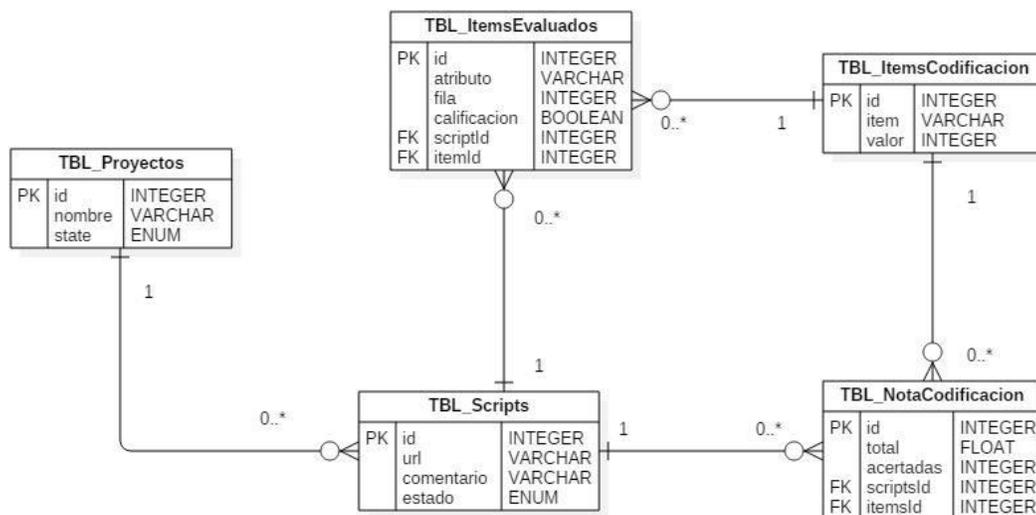


Figura 13. Segmento de diagrama para el módulo de estándares de codificación.

5.1.1 Descripción modelo entidad relación del modulo

Tabla	Descripción
TBL_Proyectos	En esta tabla se encuentran almacenados los proyectos registrados en el sistema, el cual cuenta con el id, el nombre del proyecto y el estado en que se encuentra el proyecto
TBL_Scripts	En esta tabla se encuentra almacenados archivos en el lenguaje de PHP o scripts, cuenta con un id, un url que es la ubicación de almacenamiento, un comentario dado por el evaluador y el estado en que se encuentra el script, siendo este calificado o sin calificar, su relación con la tabla "TBL_Proyectos" es de un proyecto, muchos scripts
TBL_ItemsCodificacion	En esta tabla se encuentra almacenados los ítems que van a ser calificados por el sistema, el cual el id, el ítem de codificación y el valor numérico que se le da a ese ítem
TBL_ItemsEvaluados	En esta tabla se encuentra almacenadas las calificaciones de los ítems para posteriormente generar la nota final cuenta con un id, el atributo el cual es sistema evalúa, la ubicación del atributo o la fila, la calificación del ítem, cuenta con una relación con la tabla "TBL_Scripts" el cual es de un script puede tener muchos ítems evaluados y una relación con la tabla "TBL_ItemsCodificacion" el cual muchos ítems de codificación pueden ser evaluados en la tabla "TBL_ItemsEvaluados"
TBL_NotaCodificacion	En esta tabla se encuentran almacenadas las notas finales de cada Script, es una tabla intermedia

	<p>entre “TBL_Scripts” y “TBL_ItemsCodificacion” cuenta con un id, el total de ítems de codificación encontrados y cuantas acertó de manera correcta tiene relación con la tabla “TBL_Scripts” de un script, muchas notas y tiene una relación con la tabla “TBL_ItemsCodificacion” de un ítem, muchas notas de codificación</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 2. Descripción diagrama entidad relación.

5.2 Diagramas casos de uso

Los casos de uso son una forma de describir las actividades o pasos que se deben realizar para llevar a cabo un determinado proceso. Las personas que interactúan en la aplicación se denominan actores, para este módulo se definieron tres (3) actores los cuales se describen a continuación.

5.2.1 Estudiante

Se encarga de cargar a la plataforma la documentación acerca de los scripts que componen el proyecto a evaluar, a su vez también puede eliminarlos, listarlos además puede observar la calificación obtenida por el módulo y generar reporte.

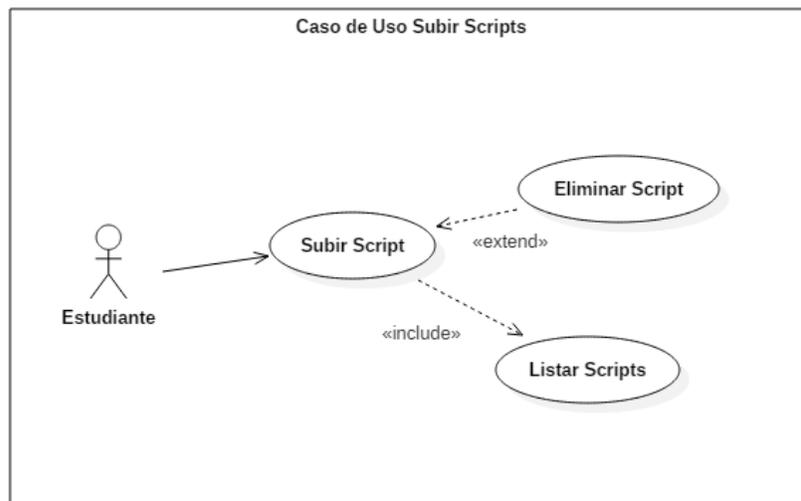


Figura 14. Caso de uso subir script.

Casos de Uso	
Nombre	Subir Script
Actores	Estudiante
Función	Subir archivo a la plataforma
Descripción	Este caso de uso permite al estudiante subir los scripts que componen el proyecto para la evaluación de cada uno de ellos.

Tabla 3. Descripción caso de uso subir script.

Casos de Uso	
Nombre	Eliminar Script
Actores	Estudiante
Función	Eliminar archivo de la plataforma
Descripción	Este caso de uso permite al estudiante eliminar los scripts que subió a la plataforma.

Tabla 4. Descripción caso de eliminar script.

Casos de Uso	
Nombre	Subir Script, Listar Scripts
Actores	Estudiante
Función	Ver listado de Scripts en plataforma
Descripción	Este caso de uso permite al estudiante ver un listado de los scripts que están almacenados en plataforma.

Tabla 5. Descripción caso de uso listar script.

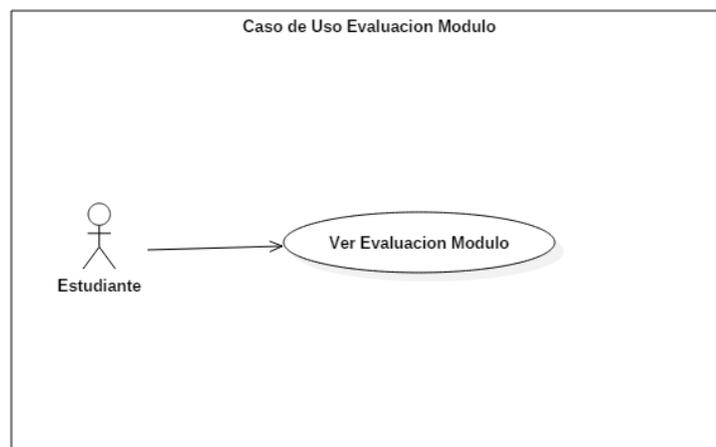


Figura 15. Caso de uso ver evaluación modulo.

Casos de Uso	
Nombre	Ver Evaluación Modulo
Actores	Estudiante
Función	Ver calificación obtenida del módulo de estándares de codificación.
Descripción	Este caso de uso permite al estudiante ver la calificación obtenida en el módulo de estándares de codificación.

Tabla 6. Descripción caso de uso ver evaluación modulo.

5.2.2 Evaluador

Se encarga de evaluar al estudiante, verificar los estándares de programación y la prioridad de cada ítem de evaluación y puede realizar observaciones sobre los documentos que ha evaluado también puede generar los reportes sobre los proyectos que evaluó.

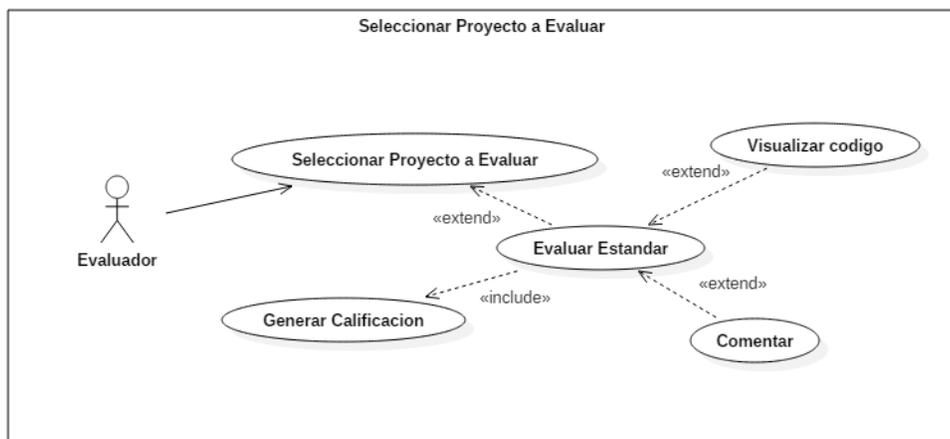


Figura 16. Caso de uso seleccionar proyecto a evaluar.

Casos de Uso	
Nombre	Seleccionar Proyecto a Evaluar
Actores	Evaluador
Función	Selecciona el proyecto que se va a evaluar
Descripción	Este caso de uso permite al evaluador seleccionar el proyecto que va evaluar de los que le hayan sido asignados.

Tabla 7. Descripción caso de uso seleccionar proyecto a evaluar.

Casos de Uso	
Nombre	Evaluar estándar
Actores	Evaluador
Función	Evalúa los scripts para verificar que se utilizó el estándar.
Descripción	Este caso de uso permite al evaluador verificar en cada uno de los scripts que componen el proyecto si se utilizó el estándar establecido.

Tabla 8. Descripción caso de uso evaluar estándar.

Casos de Uso	
Nombre	Visualizar Código.
Actores	Evaluador
Función	Genera una vista del código fuente.
Descripción	Este caso de uso permite al evaluador ver el código de fuente como en un editor de texto.

Tabla 9. Descripción caso de uso visualizar código.

Casos de Uso	
Nombre	Comentar
Actores	Evaluador
Función	Permite realizar comentarios.
Descripción	Este caso de uso permite al evaluador hacer comentarios o sugerencias sobre el proyecto.

Tabla 10. Descripción caso de uso comentar.

Casos de Uso	
Nombre	Generar Calificación.
Actores	Evaluador
Función	Genera una calificación sobre el proyecto.
Descripción	Este caso de uso permite generar una calificación en base a la revisión de los scripts.

Tabla 11. Descripción caso de uso generar calificación.

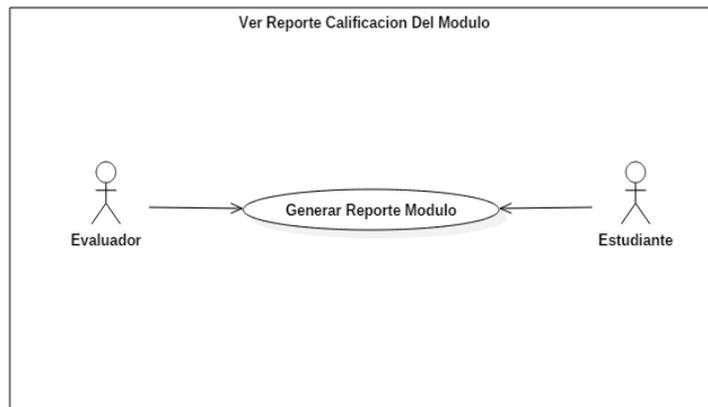


Figura 17. Caso de uso ver reporte calificación modulo.

Casos de Uso	
Nombre	Generar Reporte Modulo
Actores	Evaluador, Estudiante
Función	Genera reporte de calificación del modulo
Descripción	Este caso de uso permite al evaluador y al estudiante ver el reporte de calificación del módulo de acuerdo a su proyecto.

Tabla 12. Descripción caso de uso ver reporte de calificación modulo.

5.2.3 Administrador

Se encarga de cambiar los ítems de evaluación y manejo general de la plataforma.

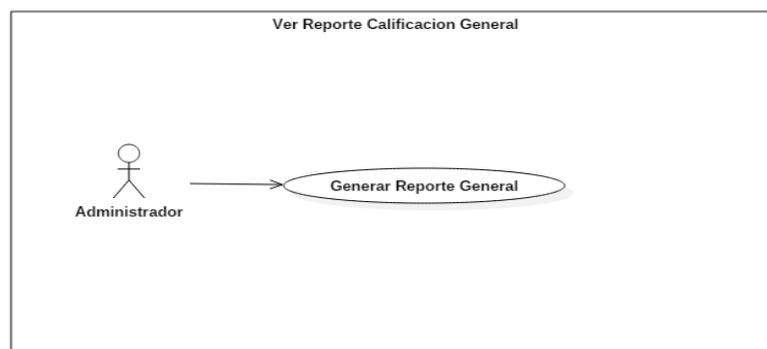


Figura 18. Caso de uso ver reporte de calificación general.

Casos de Uso	
Nombre	Generar Reporte General
Actores	Administrador
Función	Genera reporte general del modulo
Descripción	Este caso de uso permite al administrador generar reportes estadísticos sobre los proyectos evaluados en el módulo.

Tabla 13. Descripción caso de uso ver reporte de calificación general.

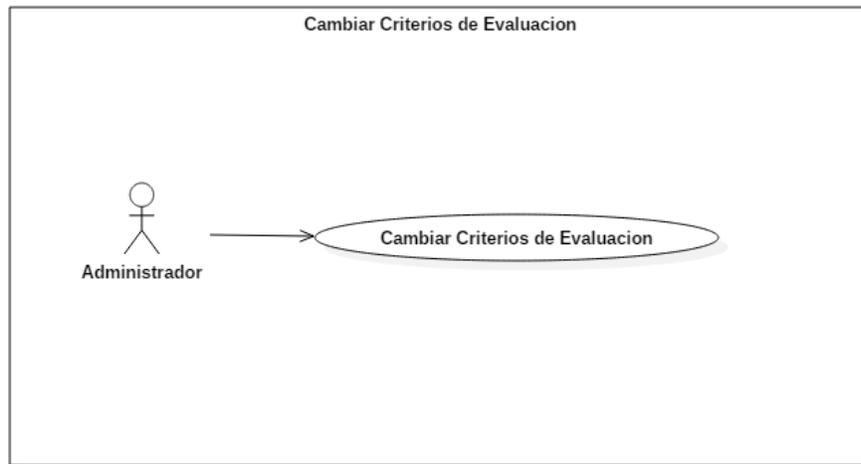


Figura 19. Caso de uso cambiar criterio de evaluación.

Casos de Uso	
Nombre	Cambiar Criterios de Evaluación
Actores	Administrador
Función	Cambia criterios y porcentajes de evaluación.
Descripción	Este caso de uso permite al administrador cambiar los porcentajes para cada una de las directrices de evaluación del módulo.

Tabla 14. Descripción caso de uso cambiar criterio de evaluación.

5.3 Diagramas de secuencia

El diagrama de secuencia es utilizado para representar la interacción entre los objetos que componen un sistema, se suelen modelar para cada caso de uso.

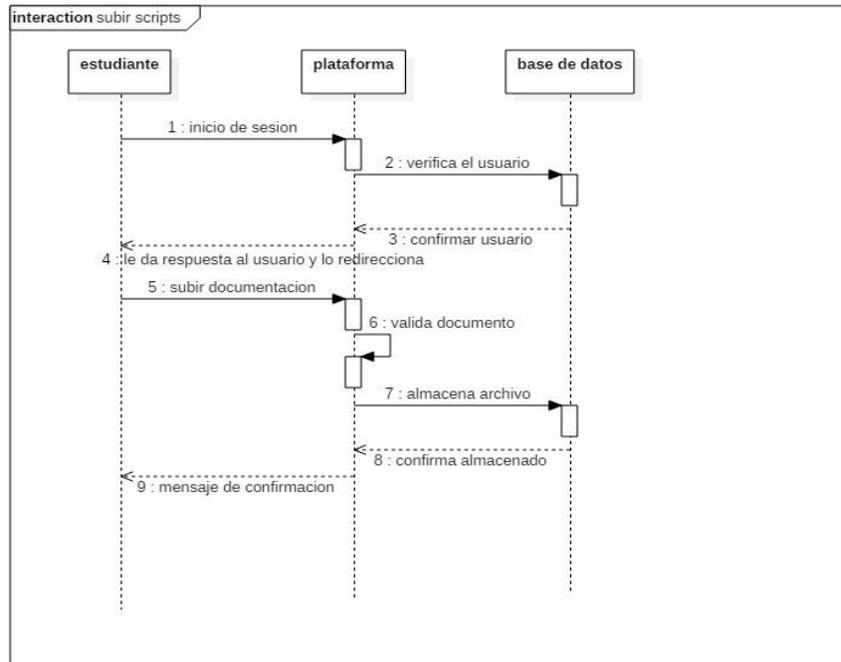


Figura 20. Diagrama de secuencia subir scripts.

Objeto	Descripción
Estudiante	El estudiante tiene una secuencia de actividades el cual su primer paso es iniciar sesión, después de la verificación y respuesta el estudiante procede a subir la documentación y su validación se le envía un mensaje de confirmación o rechazo al estudiante
Plataforma	La plataforma pasa a una secuencia de actividades las cuáles son enviar una petición a la base de datos para verificar el usuario y brindándole respuesta al estudiante, luego de subir la documentación del estudiante la plataforma realiza su validación y almacena la ruta del archivo en base de datos, luego de tener respuesta envía un mensaje de confirmación al estudiante
Base de datos	La base de datos tiene una secuencia de datos el cual luego de verificar el usuario y realizado la validación respectiva del documento script, procede a almacenarlo en base de datos y enviando la confirmación a la plataforma

Tabla 15. Descripción diagrama de secuencia subir scripts.

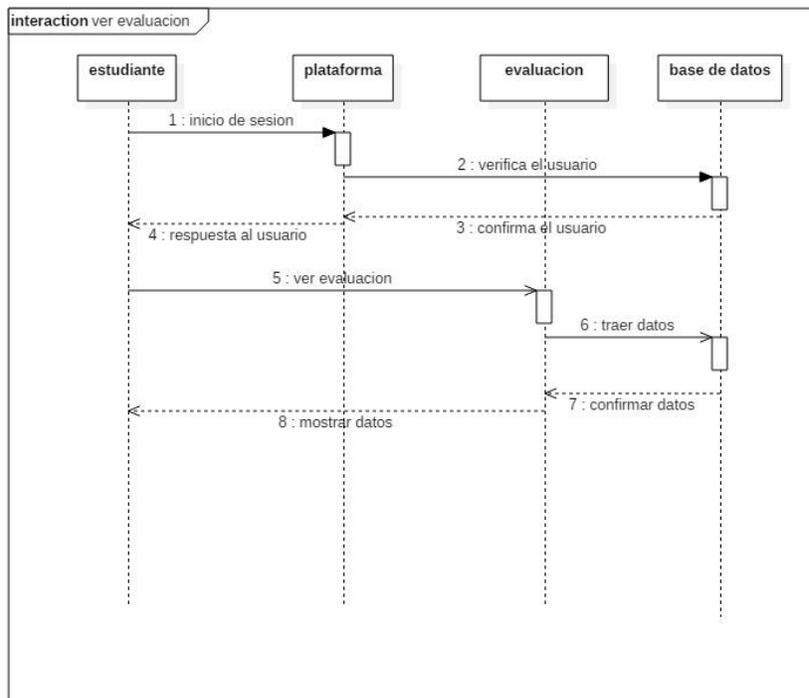


Figura 21. Diagrama de secuencia ver evaluación.

Objeto	Descripción
Estudiante	El estudiante tiene una secuencia de actividades el cual su primer paso es iniciar sesión, después de la verificación y respuesta el estudiante procede a ver la evaluación, luego de que se hallan confirmado los datos, el estudiante se le muestra los datos de evaluación
Plataforma	La plataforma pasa a una secuencia de actividades las cuáles son enviar una petición a la base de datos para verificar el usuario y brindándole respuesta al estudiante.
Evaluación	La evaluación tiene una secuencia de actividades el cual el estudiante solicita ver la evaluación, hace la petición a la base de datos para posteriormente mostrar los datos al estudiante
Base de datos	La base de datos tiene una secuencia de actividades el cual luego de verificar el usuario la evaluación solicita la traída de datos para después enviar una respuesta

Tabla 16. Descripción diagrama de secuencia ver evaluación.

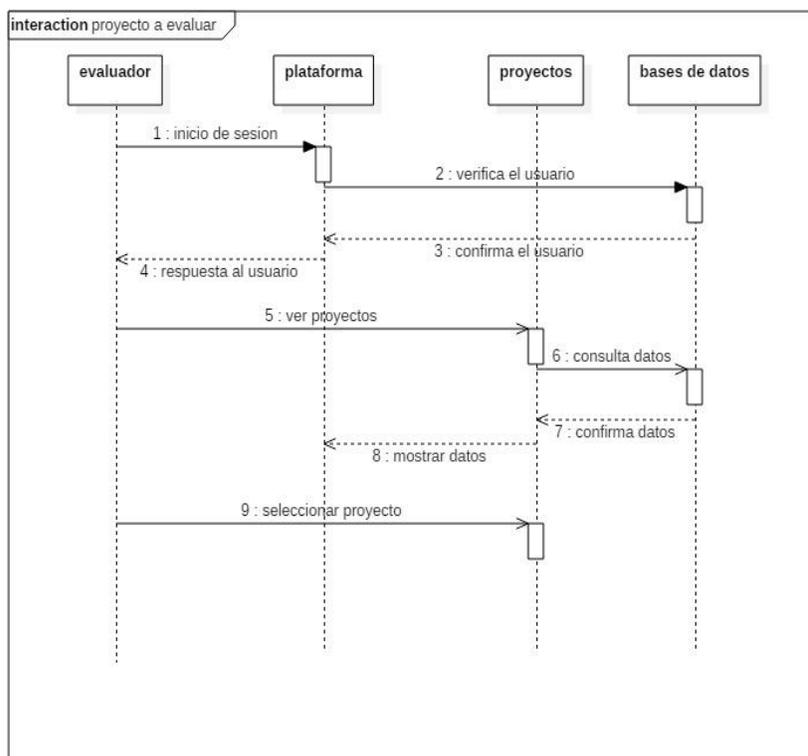


Figura 22. Diagrama de secuencia proyecto a evaluar.

Objeto	Descripción
Evaluador	El evaluador tiene una secuencia de actividades el cual su primer paso es iniciar sesión, después de la verificación y respuesta el evaluador procede a ver los proyectos, luego de que se hallan confirmado los datos, el evaluador selecciona el proyecto
Plataforma	La plataforma pasa a una secuencia de actividades las cuáles son enviar una petición a la base de datos para verificar el usuario y brindándole respuesta al evaluador.
Proyectos	Los proyectos tienen una secuencia de actividades el cual el evaluador solicita ver los proyectos de evaluación, hace la petición a la base de datos para posteriormente mostrar los datos en plataforma al evaluador
Base de datos	La base de datos tiene una secuencia de actividades el cual luego de verificar el usuario el proyecto solicita la traída de datos para después enviar una respuesta

Tabla 17. Descripción diagrama de secuencia proyecto a evaluar.

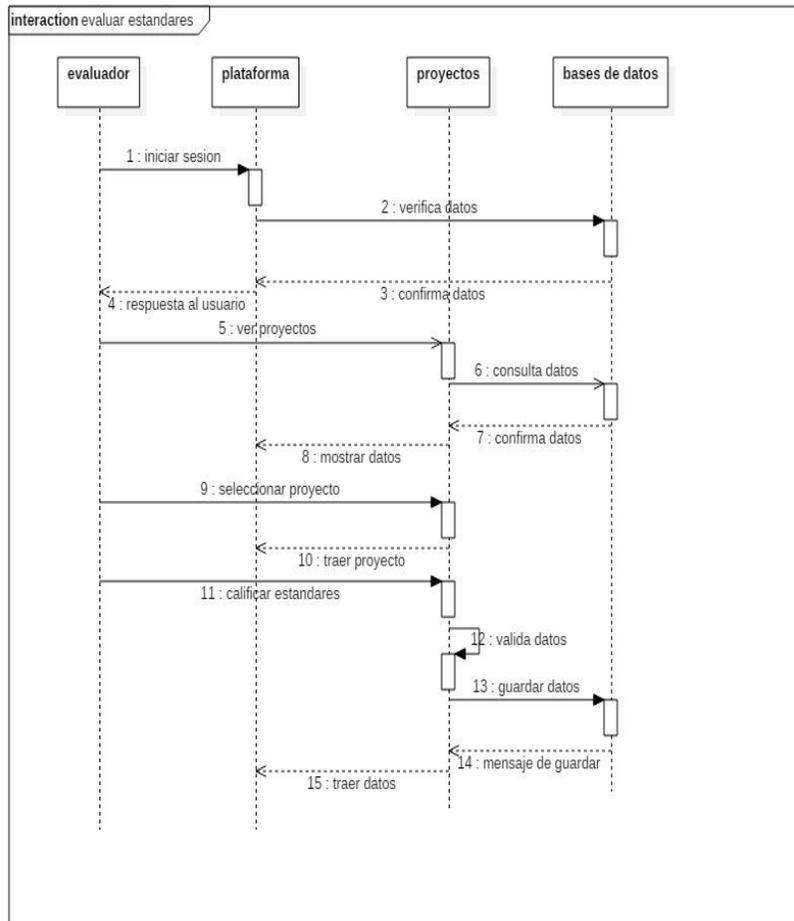


Figura 23. Diagrama de secuencia evaluar estándar.

Objeto	Descripción
Evaluador	El evaluador tiene una secuencia de actividades el cual su primer paso es iniciar sesión, después de la verificación y respuesta el evaluador procede a ver los proyectos, luego de que se hallan confirmado los datos, el evaluador selecciona el proyecto, luego que se haya traído los datos del proyecto el evaluador procede a calificar los estándares
Plataforma	La plataforma pasa a una secuencia de actividades las cuáles son enviar una petición a la base de datos para verificar el usuario y brindándole respuesta al evaluador, luego que el evaluador seleccione el proyecto, la petición de la evaluación hace la traída de datos
Proyectos	Los proyectos tienen una secuencia de actividades el cual el evaluador solicita ver los proyectos de evaluación, hace la petición a la base de datos para posteriormente mostrar los datos en plataforma al evaluador, luego que el evaluador procede a calificar los estándares y se hallan verificado hace la petición a la base de datos para la traída de información
Base de datos	La base de datos tiene una secuencia de actividades el cual luego de verificar el usuario el proyecto solicita la traída de datos para después enviar una respuesta, posteriormente se validan los datos y se hacen el guardado de la calificación de los estándares de codificación y envía el mensaje hacia la evaluación

Tabla 18. Descripción diagrama de secuencia evaluar estándar.

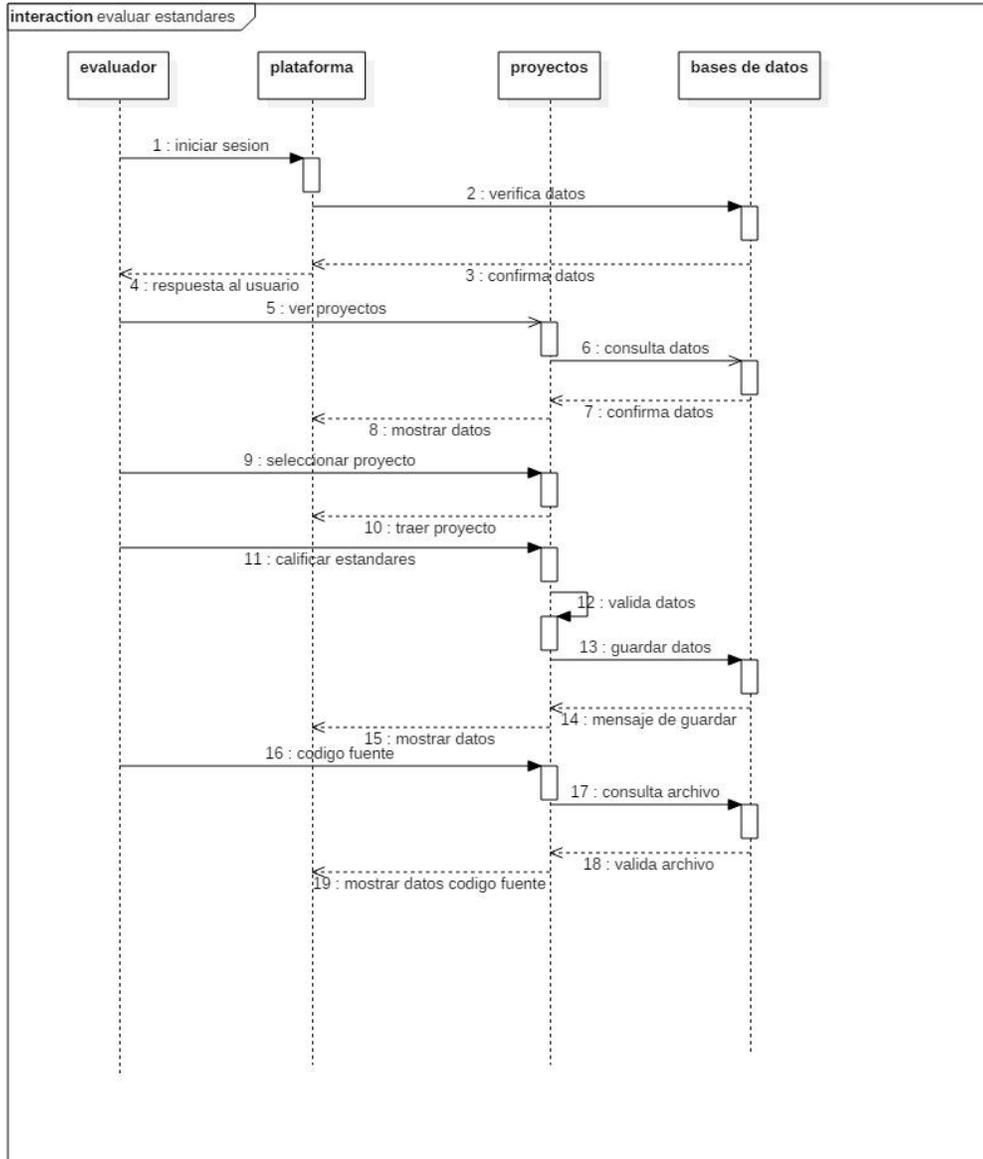


Figura 24. Diagrama de secuencia visualizar código fuente.

Objeto	Descripción
Evaluador	El evaluador tiene una secuencia de actividades el cual su primer paso es iniciar sesión, después de la verificación y respuesta el evaluador procede a ver los proyectos, luego de que se hallan confirmado los datos, el evaluador selecciona el proyecto, luego que se haya traído los datos del proyecto el evaluador procede a calificar los estándares, así mismo el evaluador envía una petición para la vista del código fuente
Plataforma	La plataforma pasa a una secuencia de actividades las cuáles son enviar una petición a la base de datos para verificar el usuario y brindándole respuesta al evaluador, luego que el evaluador seleccione el proyecto, la petición de la evaluación hace la traída de datos, así mismo luego que el evaluador realice la petición de vista del código y haber consultado en base de datos, la plataforma realiza su respectiva vista del código fuente.
Proyectos	Los proyectos tienen una secuencia de actividades el cual el evaluador solicita ver los proyectos de evaluación, hace la petición a la base de datos para posteriormente mostrar los datos en plataforma al evaluador, luego que el evaluador procede a calificar los estándares y se hallan verificado hace la petición a la base de datos para la traída de información así mismo en la vista del código fuente hace una petición a la base de datos y validando el archivo para realizar envió a la plataforma
Base de datos	La base de datos tiene una secuencia de actividades el cual luego de verificar el usuario el proyecto solicita

la traída de datos para después enviar una respuesta, posteriormente se validan los datos y se hacen el guardado de la calificación de los estándares de codificación y envía el mensaje hacia la evaluación, así mismo en la vista del código fuente hace una petición a la base de datos y validando el archivo para realizar él envió a la plataforma

Tabla 19. Descripción diagrama de secuencia visualizar código fuente.

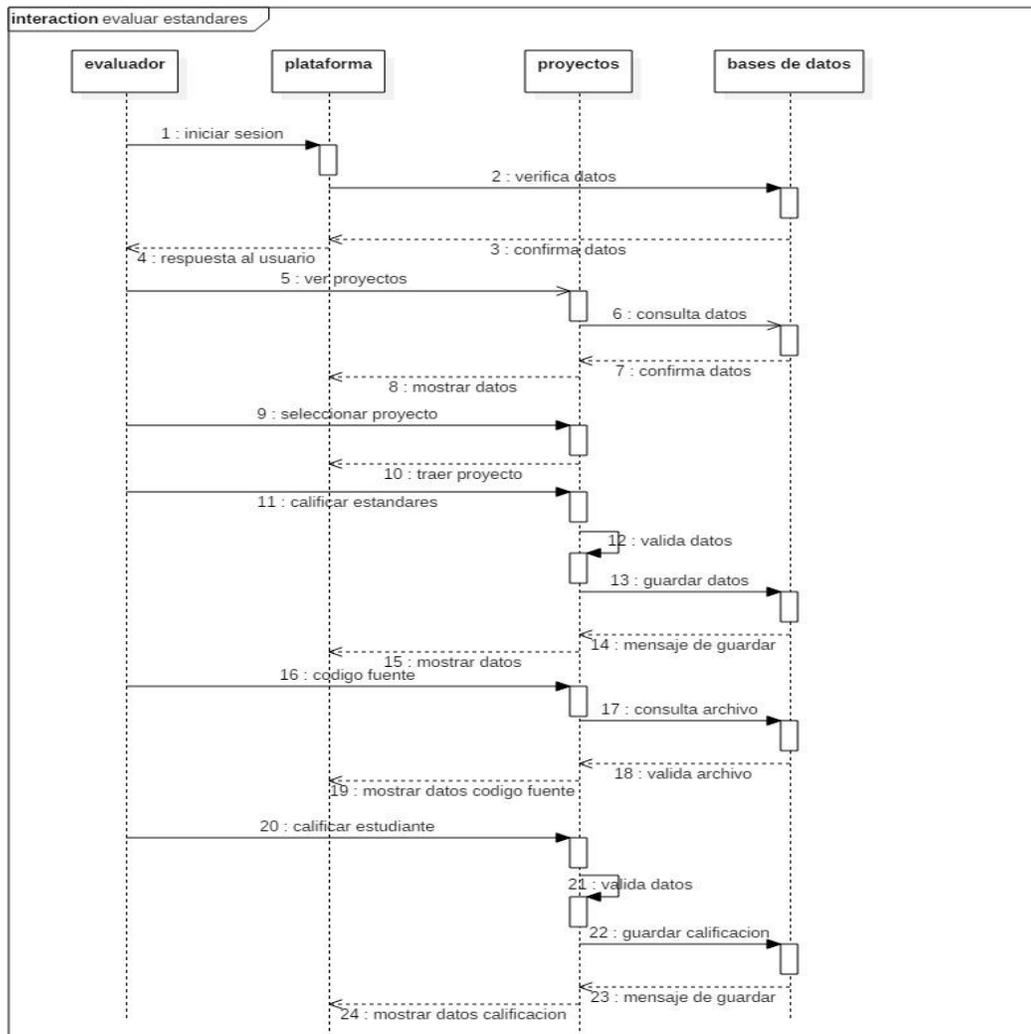


Figura 25. Diagrama de secuencia generar calificación.

Objeto	Descripción
Evaluador	El evaluador tiene una secuencia de actividades el cual su primer paso es iniciar sesión, después de la verificación y respuesta el evaluador procede a ver los proyectos, luego de que se hallan confirmado los datos, el evaluador selecciona el proyecto, luego que se haya traído los datos del proyecto el evaluador procede a calificar los estándares, así mismo el evaluador envía una petición para la vista del código fuente de esta manera el evaluador podrá calificar al estudiante enviando una petición al modulo de proyectos
Plataforma	La plataforma pasa a una secuencia de actividades las cuáles son enviar una petición a la base de datos para verificar el usuario y brindándole respuesta al evaluador, luego que el evaluador seleccione el proyecto, la petición de la evaluación hace la traída de datos, así mismo luego que el evaluador realice la petición de vista del código y haber consultado en base de datos, la plataforma realiza su respectiva vista del código fuente, de esta manera la plataforma muestra los datos de calificación final
Proyectos	Los proyectos tienen una secuencia de actividades el cual el evaluador solicita ver los proyectos de evaluación, hace la petición a la base de datos para posteriormente mostrar los datos en plataforma al evaluador, luego que el evaluador procede a calificar los estándares y se hallan verificado hace la petición a la base de datos para la traída de información así mismo en la vista del código fuente hace una petición a la base de datos y validando el archivo para realizar el envío a la plataforma , de

	<p>esta manera cuando se recibe la petición de calificación de estudiante realiza, genera los parámetros para guardar y validar la calificación consultando a base de datos</p>
Base de datos	<p>La base de datos tiene una secuencia de actividades el cual luego de verificar el usuario el proyecto solicita la traída de datos para después enviar una respuesta, posteriormente se validan los datos y se hacen el guardado de la calificación de los estándares de codificación y envía el mensaje hacia la evaluación, así mismo en la vista del código fuente hace una petición a la base de datos y validando el archivo para realizar él envió a la plataforma, de esta manera cuando se recibe una petición de calificar al estudiante procede a guardar los datos y luego de ser consultados envía los datos a plataforma</p>

Tabla 20. Descripción diagrama de secuencia generar calificación.

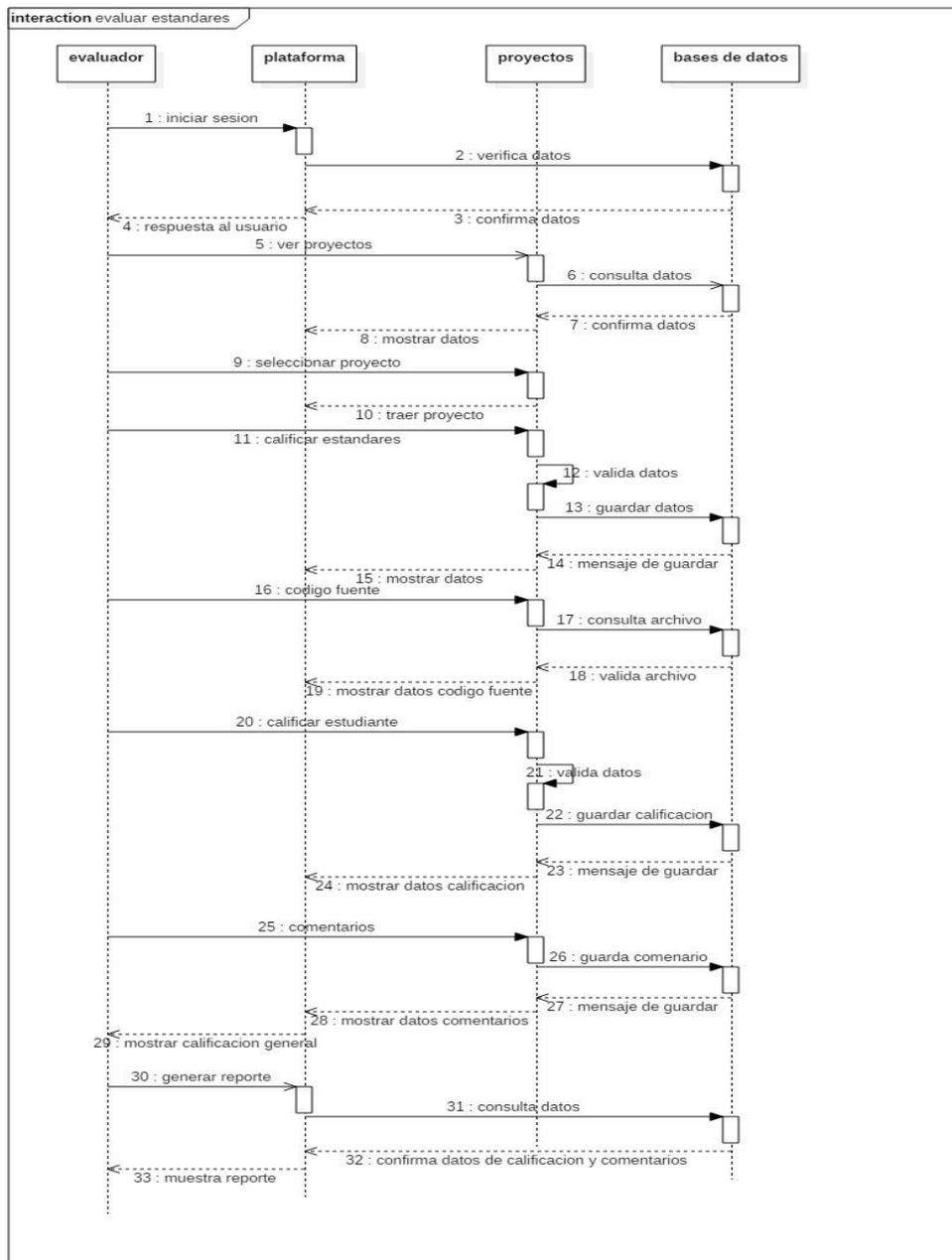


Figura 26. Diagrama de secuencia ver reporte de calificación.

Objeto	Descripción
Evaluador	El evaluador tiene una secuencia de actividades el cual su primer paso es iniciar sesión, después de la verificación y respuesta el evaluador procede a ver los proyectos, luego de que se hallan confirmado los datos, el evaluador selecciona el proyecto, luego que se haya traído los datos del proyecto el evaluador procede a calificar los estándares, así mismo el evaluador envía una petición para la vista del código fuente de esta manera el evaluador podrá calificar al estudiante enviando una petición al módulo de proyectos, por ultimo luego de proceder a que el evaluador realice o no un comentario de la calificación, el evaluador genera el reporte general realizando la petición directamente a base de datos y después de su debida confirmación procede a la muestra de la calificación final
Plataforma	La plataforma pasa a una secuencia de actividades las cuáles son enviar una petición a la base de datos para verificar el usuario y brindándole respuesta al evaluador, luego que el evaluador seleccione el proyecto, la petición de la evaluación hace la traída de datos, así mismo luego que el evaluador realice la petición de vista del código y haber consultado en base de datos, la plataforma realiza su respectiva vista del código fuente, de esta manera la plataforma muestra los datos de calificación final finalmente cuando el evaluador hace la petición de comentario y luego de proceder a guardarlo realiza su respectiva muestra y así generar el

	<p>reporte final enviando la petición a la base de datos y luego de su respuesta envía al evaluador el reporte final</p>
Proyectos	<p>Los proyectos tienen una secuencia de actividades el cual el evaluador solicita ver los proyectos de evaluación, hace la petición a la base de datos para posteriormente mostrar los datos en plataforma al evaluador, luego que el evaluador procede a calificar los estándares y se hallan verificado hace la petición a la base de datos para la traída de información así mismo en la vista del código fuente hace una petición a la base de datos y validando el archivo para realizar el envío a la plataforma , de esta manera cuando se recibe la petición de calificación de estudiante realiza, genera los parámetros para guardar y validar la calificación consultando a base de datos</p>
Base de datos	<p>La base de datos tiene una secuencia de actividades el cual luego de verificar el usuario el proyecto solicita la traída de datos para después enviar una respuesta, posteriormente se validan los datos y se hacen el guardado de la calificación de los estándares de codificación y envía el mensaje hacia la evaluación, así mismo en la vista del código fuente hace una petición a la base de datos y validando el archivo para realizar él envío a la plataforma, de esta manera cuando se recibe una petición de calificar al estudiante procede a guardar los datos y luego de ser consultados envía los datos a plataforma, finalmente luego de haber guardado la petición del comentario, se realiza una consulta para enviarle a la plataforma el</p>

reporte para la respectiva vista de las notas finales

Tabla 21. Descripción diagrama de secuencia ver reporte de calificación.

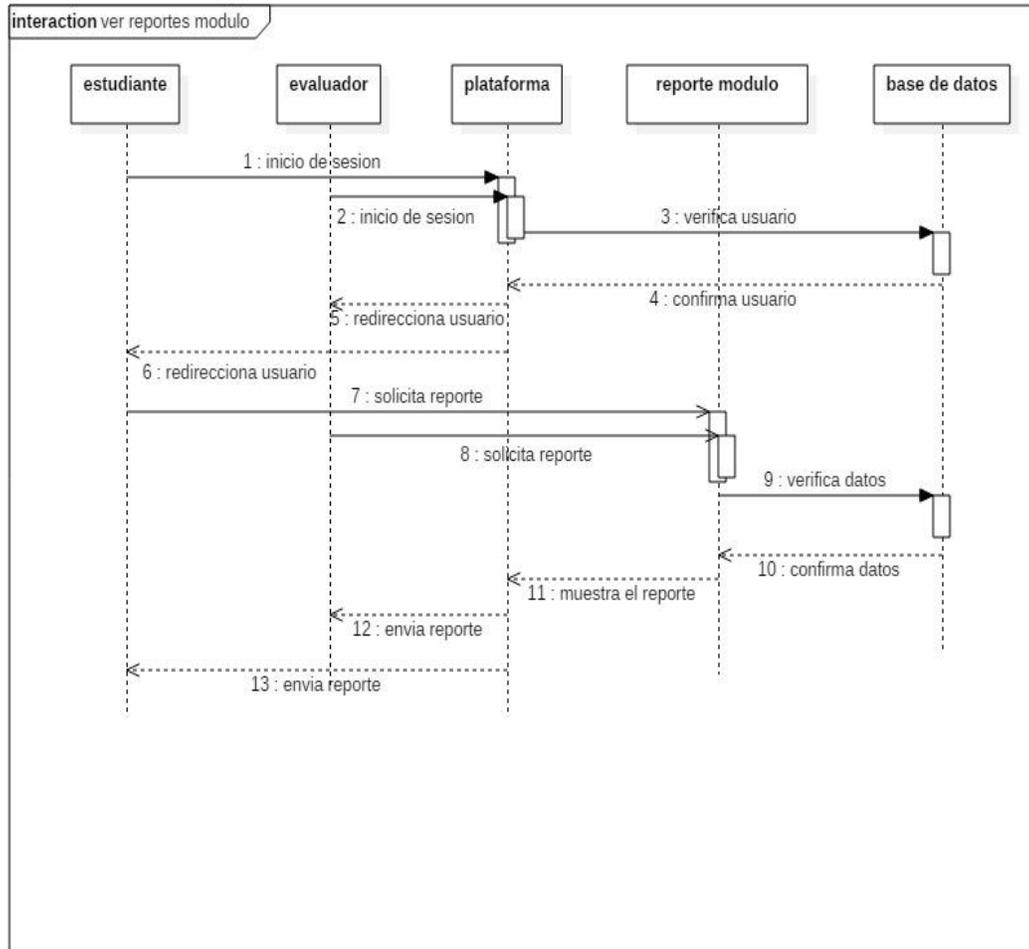


Figura 27. Diagrama de secuencia ver reporte general del módulo.

Objeto	Descripción
Estudiante/Evaluador	El estudiante y evaluador tiene una secuencia de actividades el cual su primer paso es iniciar sesión, después de la verificación y respuesta el estudiante o docente procede solicitar el reporte del módulo para luego realizar la muestra de datos generales
Plataforma	La plataforma pasa a una secuencia de actividades las cuáles son enviar una petición a la base de datos para verificar el usuario y brindándole respuesta al estudiante o evaluador
Reporte del modulo	El reporte del módulo tiene una secuencia de actividades el cual el estudiante o evaluador solicita ver la evaluación general del todos los módulos , hace la petición a la base de datos para posteriormente mostrar los datos al usuario
Base de datos	La base de datos tiene una secuencia de actividades el cual luego de verificar el usuario recibe la petición del reporte del módulo lo cual verifica y realiza el envío respectivo de los datos para el reporte final del modulo

Tabla 22. Descripción diagrama de secuencia ver reporte general del módulo.

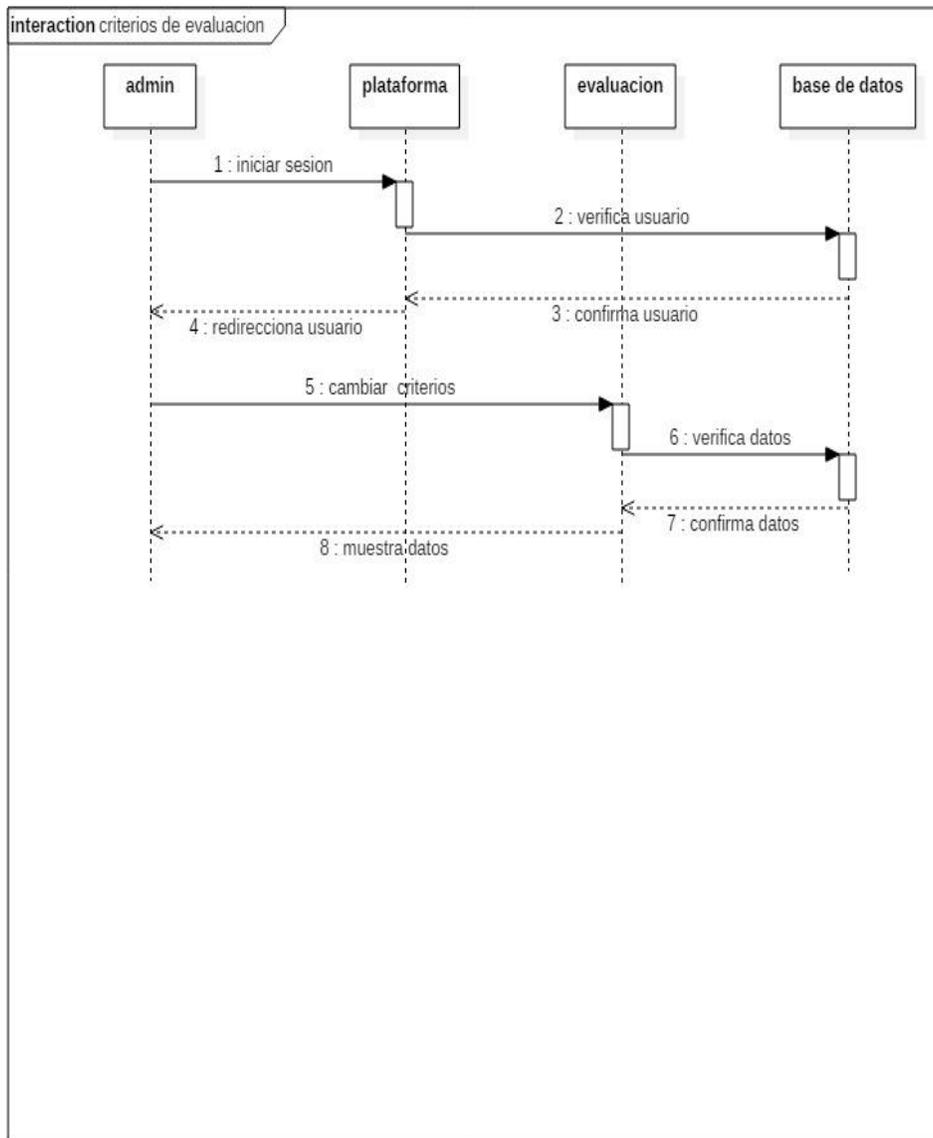


Figura 28. Diagrama de secuencia cambio criterios de evaluación.

Objeto	Descripción
Administrador	El administrador tiene una secuencia de actividades el cual su primer paso es iniciar sesión, después de la verificación y respuesta el evaluador procede a cambiar los criterios de evaluación, finalmente recibe respuesta del módulo de evaluación
Plataforma	La plataforma pasa a una secuencia de actividades las cuáles son enviar una petición a la base de datos para verificar el usuario y brindándole respuesta al evaluador.
Evaluación	La evaluación tiene una secuencia de actividades el cual cuando el administrador envía la petición de cambios de criterios solicita los datos a la base de datos y luego de la respuesta procede a la muestra de datos al administrador
Base de datos	La base de datos tiene una secuencia de datos el cual luego de verificar el usuario y realizado la validación respectiva de verificación de cambio de criterios procede a almacenarlo en base de datos y enviando la confirmación al módulo de evaluación

Tabla 23. Descripción diagrama de secuencia cambio criterios de evaluación.

5.4 Diagramas de actividades

El diagrama de actividades muestra un proceso de software como un flujo de trabajo a través de una serie de acciones, que pueden ser realizadas por personas, componentes u otros equipos.

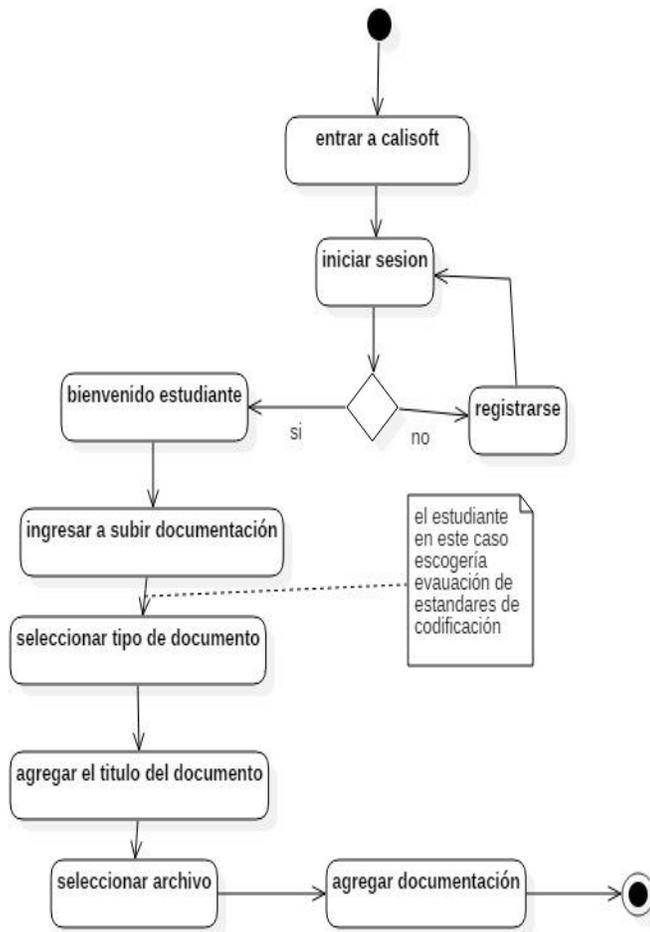


Figura 29. Diagrama de actividades subir script.

Actividades	Descripción
Entrar a Calisoft	Inicio de la plataforma de Calisoft
Iniciar sesión	El usuario procede a acceder sus datos a el inicio de sesión
Bienvenido estudiante	Si el estudiante inicia sesión de manera exitosa procede a mostrar la página de inicio del estudiante
Registrarse	El es estudiante inicia sesión de manera incorrecta, se muestra el formulario de registro del sistema
Ingresar a subir documentación	Si el estudiante está avalado por el administrador y dado su evaluador, el estudiante puede proceder a entrar a subir la documentación
Seleccionar el tipo de documento	El estudiante en este caso escogería el módulo para la evaluación de estándares de codificación
Agregar el título de documento	El estudiante procede a ingresar al módulo para la evaluación de estándares de codificación
Seleccionar archivo	El estudiante procede a seleccionar los scripts o archivos de codificación en PHP
Agregar documentación	Si el estudiante realiza todo de manera exitosa puede agregar la documentación del módulo para la evaluación de estándares
Cerrar sesión	Una vez cumplida cualquiera de estas tareas el estudiante podrá cerrar sesión si lo desea

Tabla 24. Descripción diagrama de actividades subir script.

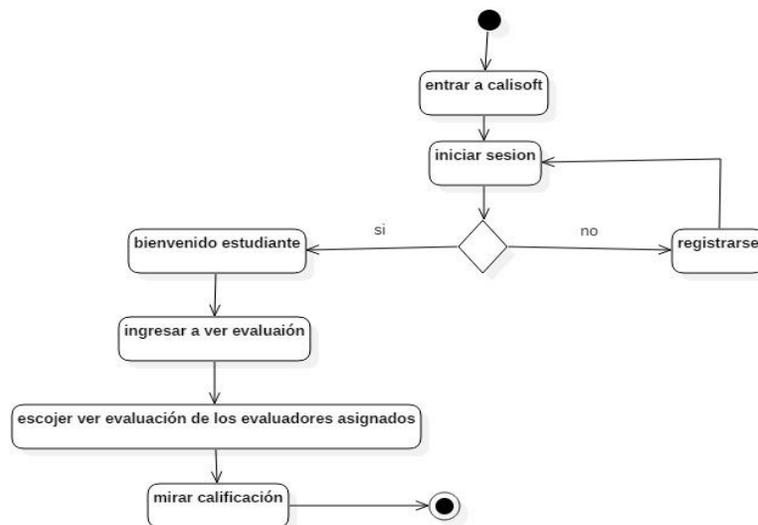


Figura 30. Diagrama de actividades ver evaluación del módulo.

Actividades	Descripción
Entrar a Calisoft	Inicio de la plataforma de Calisoft
Iniciar sesión	El usuario procede a acceder sus datos a el inicio de sesión
Bienvenido estudiante	Si el estudiante inicia sesión de manera exitosa procede a mostrar la página de inicio del estudiante
Registrarse	El estudiante inicia sesión de manera incorrecta, se muestra el formulario de registro del sistema
Ingresar a ver evaluación	Si el evaluador califica de manera autónoma los archivos de codificación el estudiante se habilita a ver evaluación
Escoger ver evaluación de los evaluadores asignados	El estudiante en este caso escogería ver evaluación de los evaluadores asignados
Mirar calificación	Si se realiza todo de manera exitosa podrá mirar el reporte final del módulo para la evaluación de estándares de codificación
Cerrar sesión	Una vez cumplida cualquiera de estas tareas el estudiante podrá cerrar sesión si lo desea

Tabla 25. Descripción diagrama de actividades ver evaluación del módulo.

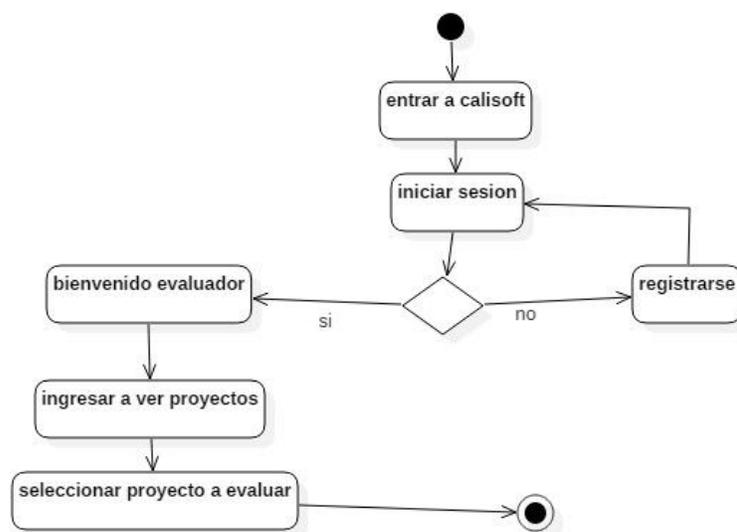


Figura 31. Diagrama de actividades seleccionar proyecto a evaluar.

Actividades	Descripción
Entrar a Calisoft	Inicio de la plataforma de Calisoft
Iniciar sesión	El usuario procede a acceder sus datos a el inicio de sesión
Bienvenido evaluador	Si el evaluador inicia sesión de manera exitosa procede a mostrar la página de inicio del evaluador
Registrarse	El evaluador inicia sesión de manera incorrecta, solicitar restaurar contraseña
Ingresar a ver proyectos	El evaluador accede a ver los proyectos que se le han asignado por parte del administrador
Seleccionar proyecto a evaluar	El evaluador selecciona proyecto asignado por parte del administrador
Cerrar sesión	Una vez cumplida cualquiera de estas tareas el evaluador podrá cerrar sesión si lo desea

Tabla 26. Descripción diagrama de actividades seleccionar proyecto a evaluar.

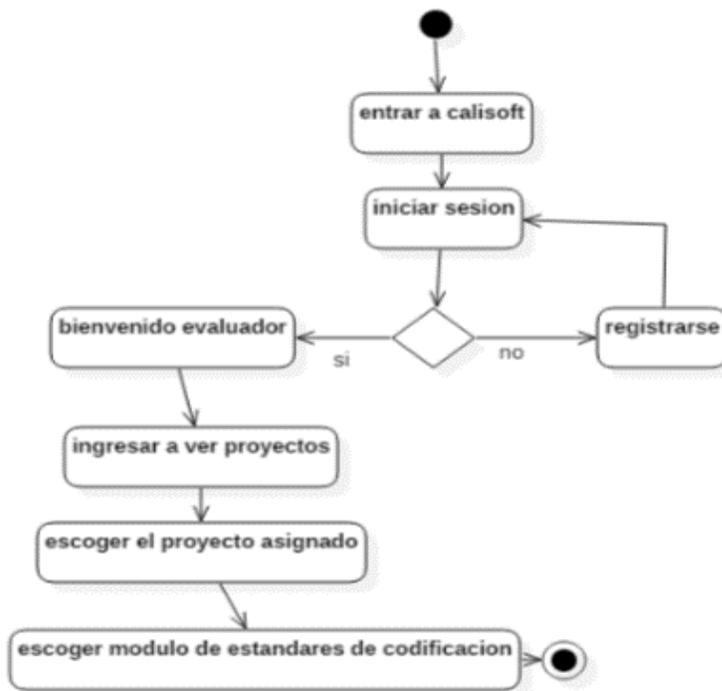


Figura 32. Diagrama de actividades escoger script de codificación.

Actividades	Descripción
Entrar a Calisoft	Inicio de la plataforma de Calisoft
Iniciar sesión	El usuario procede a acceder sus datos a el inicio de sesión
Bienvenido evaluador	Si el evaluador inicia sesión de manera exitosa procede a mostrar la página de inicio del evaluador
Registrarse	El evaluador inicia sesión de manera incorrecta, solicitar restaurar contraseña
Ingresar a ver proyectos	El evaluador accede a ver los proyectos que se le han asignado por parte del administrador
Escoger módulo de estándares de codificación	El evaluador selecciona el módulo de codificación
Cerrar sesión	Una vez cumplida cualquiera de estas tareas el evaluador podrá cerrar sesión si lo desea

Tabla 27. Diagrama de actividades escoger script de codificación.

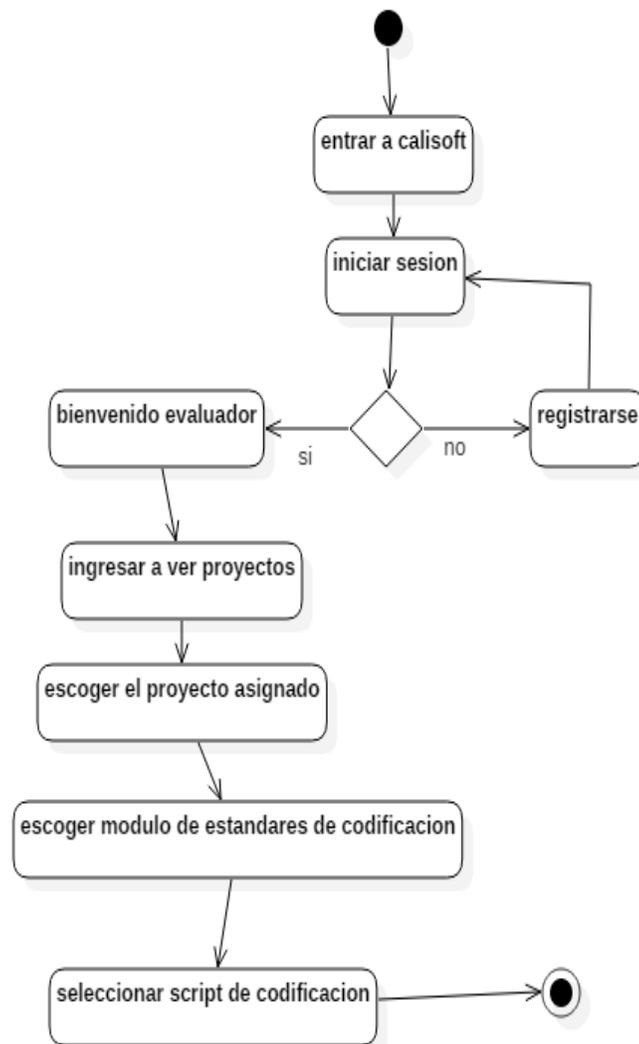


Figura 33. Diagrama de actividades seleccionar script.

Actividades	Descripción
Entrar a Calisoft	Inicio de la plataforma de Calisoft
Iniciar sesión	El usuario procede a acceder sus datos a el inicio de sesión
Bienvenido evaluador	Si el evaluador inicia sesión de manera exitosa procede a mostrar la página de inicio del evaluador
Registrarse	El evaluador inicia sesión de manera incorrecta, solicitar restaurar contraseña
Ingresar a ver proyectos	El evaluador accede a ver los proyectos que se le han asignado por parte del administrador
Escoger módulo de estándares de codificación	El evaluador selecciona el módulo de codificación
Seleccionar Script de codificación	El evaluador selecciona un script para proceder a realizar la calificación
Cerrar sesión	Una vez cumplida cualquiera de estas tareas el evaluador podrá cerrar sesión si lo desea

Tabla 28. Descripción diagrama de actividades seleccionar script.

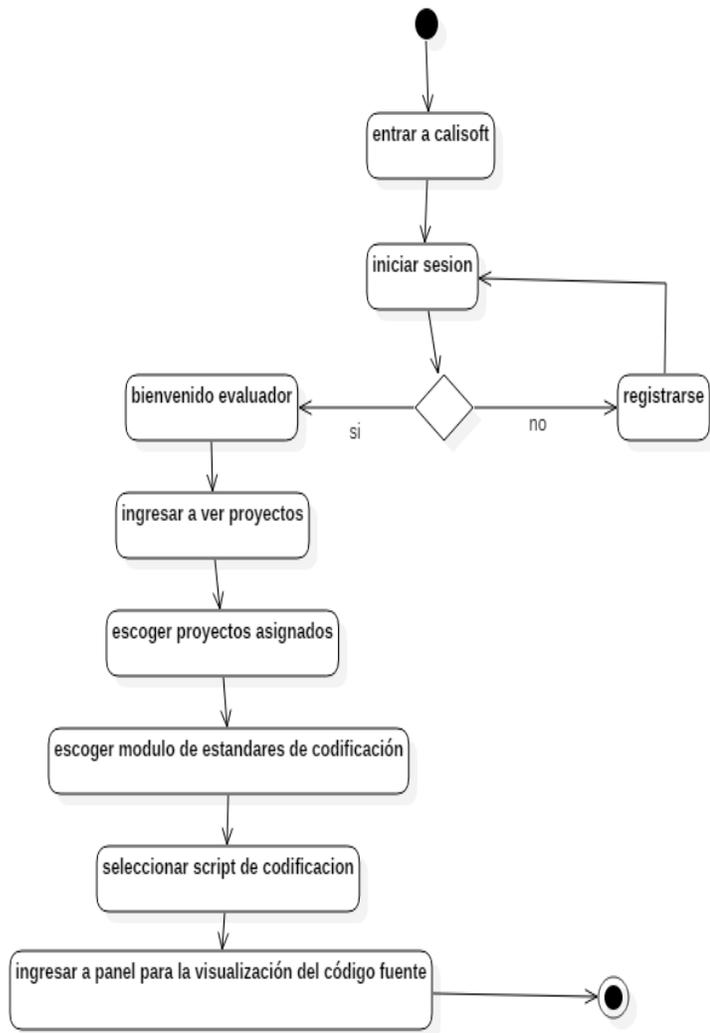


Figura 34. Diagrama de actividades visualizar código fuente.

Actividades	Descripción
Entrar a Calisoft	Inicio de la plataforma de Calisoft
Iniciar sesión	El usuario procede a acceder sus datos a el inicio de sesión
Bienvenido evaluador	Si el evaluador inicia sesión de manera exitosa procede a mostrar la página de inicio del evaluador
Registrarse	El evaluador inicia sesión de manera incorrecta, solicitar restaurar contraseña
Ingresar a ver proyectos	El evaluador accede a ver los proyectos que se le han asignado por parte del administrador
Escoger módulo de estándares de codificación	El evaluador selecciona el módulo de codificación
Seleccionar Script de codificación	El evaluador selecciona un script para proceder a realizar la calificación
Ingresar al panel para la visualización del código fuente	Se muestra el código fuente del script seleccionado con los parámetros de calificación
Cerrar sesión	Una vez cumplida cualquiera de estas tareas el evaluador podrá cerrar sesión si lo desea

Tabla 29. Descripción diagrama de actividades visualizar código fuente.

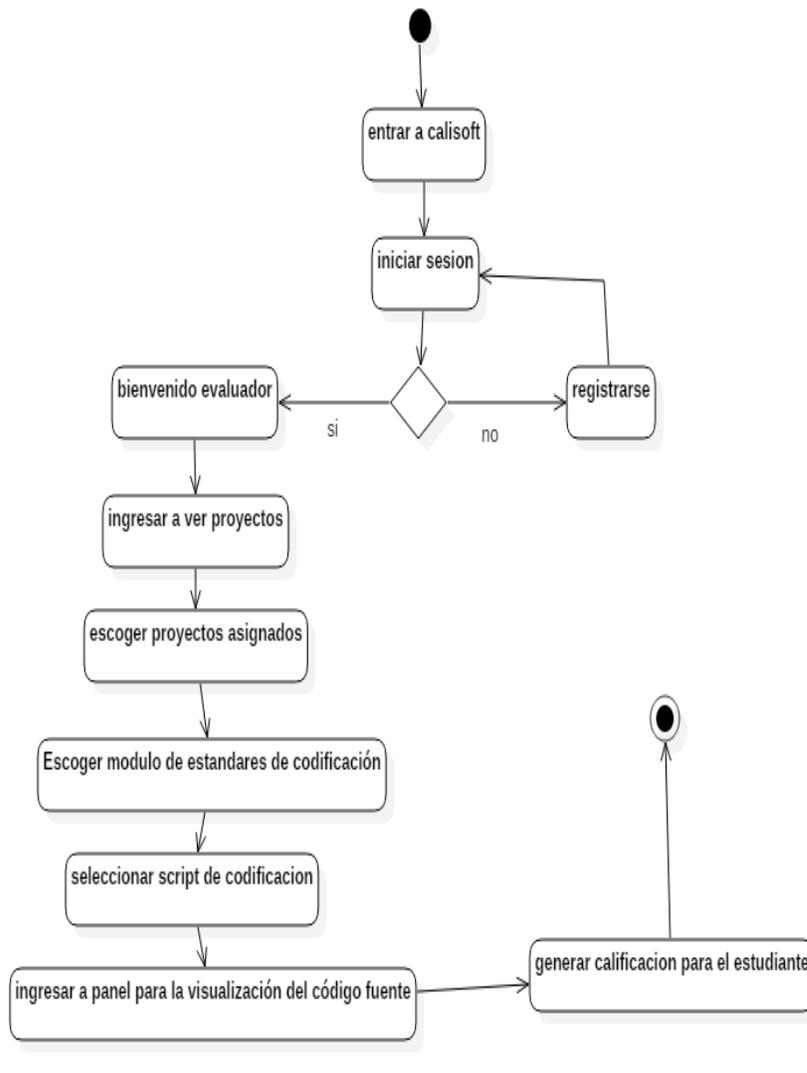


Figura 35. Diagrama de actividades generar calificación.

Actividades	Descripción
Entrar a Calisoft	Inicio de la plataforma de Calisoft
Iniciar sesión	El usuario procede a acceder sus datos a el inicio de sesión
Bienvenido evaluador	Si el evaluador inicia sesión de manera exitosa procede a mostrar la página de inicio del evaluador
Registrarse	El evaluador inicia sesión de manera incorrecta, solicitar restaurar contraseña
Ingresar a ver proyectos	El evaluador accede a ver los proyectos que se le han asignado por parte del administrador
Escoger módulo de estándares de codificación	El evaluador selecciona el módulo de codificación
Seleccionar Script de codificación	El evaluador selecciona un script para proceder a realizar la calificación
Ingresar al panel para la visualización del código fuente	Se muestra el código fuente del script seleccionado con los parámetros de calificación
Generar calificación al estudiante	Una vez que se muestre el código, el evaluador podrá escoger si realizar evaluación automática o manual del archivo de codificación
Cerrar sesión	Una vez cumplida cualquiera de estas tareas el evaluador podrá cerrar sesión si lo desea

Tabla 30. Descripción diagrama de actividades generar calificación.

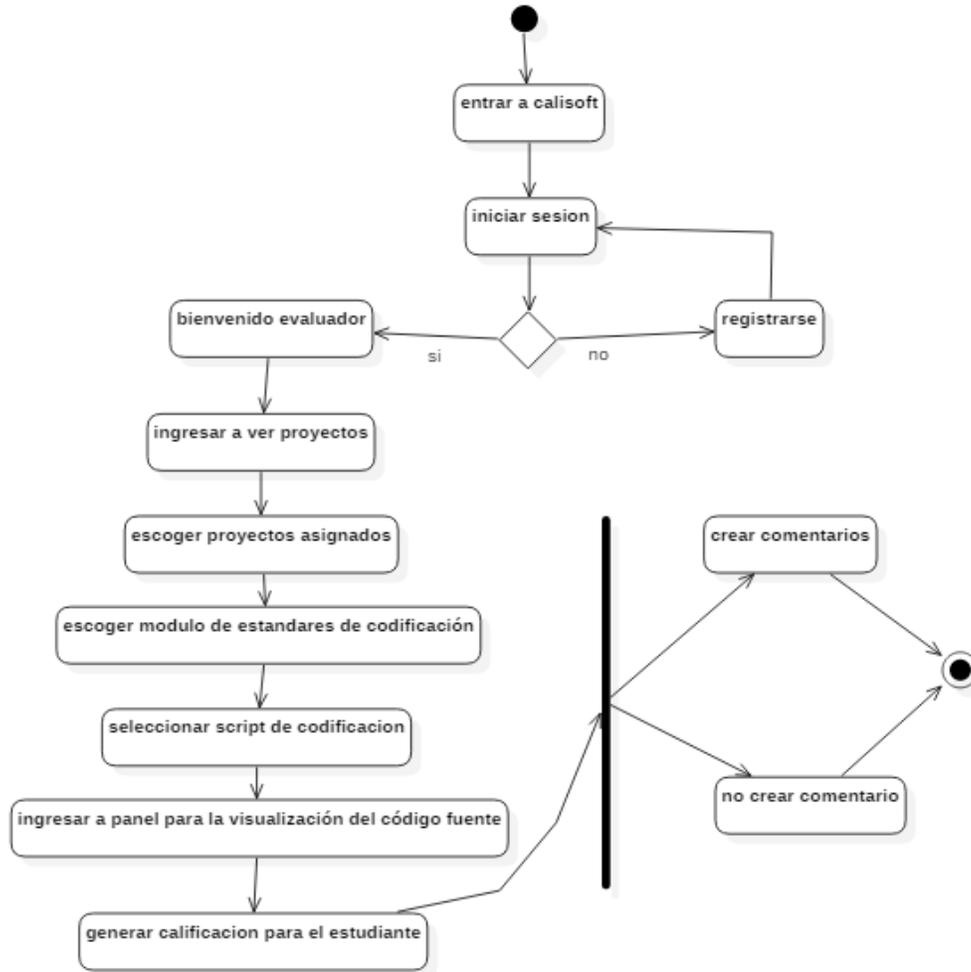


Figura 36. Diagrama de actividades generar comentario.

Actividades	Descripción
Entrar a Calisoft	Inicio de la plataforma de Calisoft
Iniciar sesión	El usuario procede a acceder sus datos a el inicio de sesión
Bienvenido evaluador	Si el evaluador inicia sesión de manera exitosa procede a mostrar la página de inicio del evaluador
Registrarse	El evaluador inicia sesión de manera incorrecta, solicitar restaurar contraseña
Ingresar a ver proyectos	El evaluador accede a ver los proyectos que se le han asignado por parte del administrador
Escoger módulo de estándares de codificación	El evaluador selecciona el módulo de codificación
Seleccionar Script de codificación	El evaluador selecciona un script para proceder a realizar la calificación
Ingresar al panel para la visualización del código fuente	Se muestra el código fuente del script seleccionado con los parámetros de calificación
Generar calificación al estudiante	Una vez que se muestre el código, el evaluador podrá escoger si realizar evaluación automática o manual del archivo de codificación
Crear comentarios	El evaluador puede escoger si crear un comentario al script calificado
No crear comentario	El evaluador puede escoger no crear un comentario al script calificado
Cerrar sesión	Una vez cumplida cualquiera de estas tareas el evaluador podrá cerrar sesión si lo desea

Tabla 31. Descripción diagrama de actividades generar comentario.

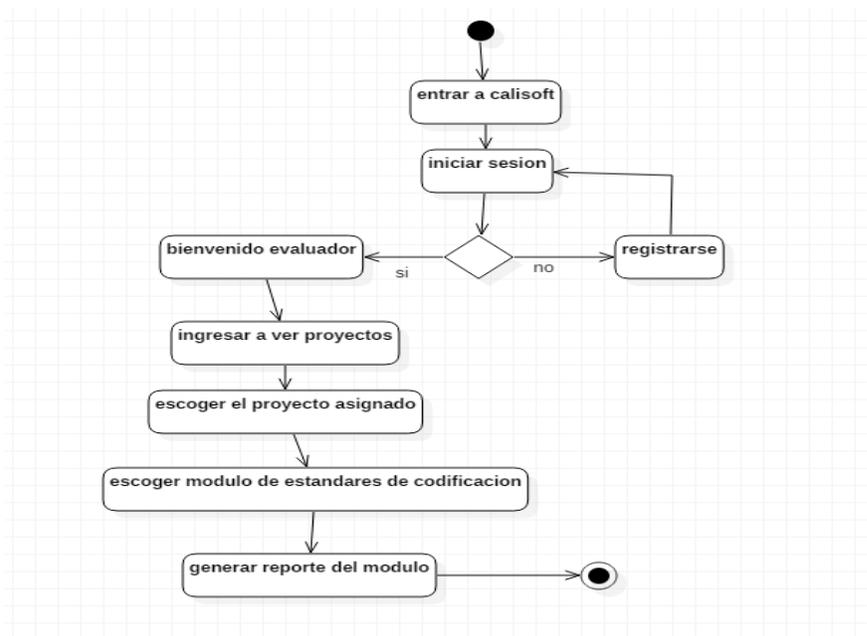


Figura 37. Diagrama de actividades generar reporte evaluador.

Actividades	Descripción
Entrar a Calisoft	Inicio de la plataforma de Calisoft
Iniciar sesión	El usuario procede a acceder sus datos a el inicio de sesión
Bienvenido evaluador	Si el evaluador inicia sesión de manera exitosa procede a mostrar la página de inicio del evaluador
Registrarse	El evaluador inicia sesión de manera incorrecta, solicitar restaurar contraseña
Ingresar a ver proyectos	El evaluador accede a ver los proyectos que se le han asignado por parte del administrador
Escoger módulo de estándares de codificación	El evaluador selecciona el módulo de codificación
Generar reporte del modulo	El evaluador puede observar el reporte del módulo de evaluación de codificación del proyecto asignado
Cerrar sesión	Una vez cumplida cualquiera de estas tareas el evaluador podrá cerrar sesión si lo desea

Tabla 32. Descripción diagrama de actividades generar reporte evaluador.

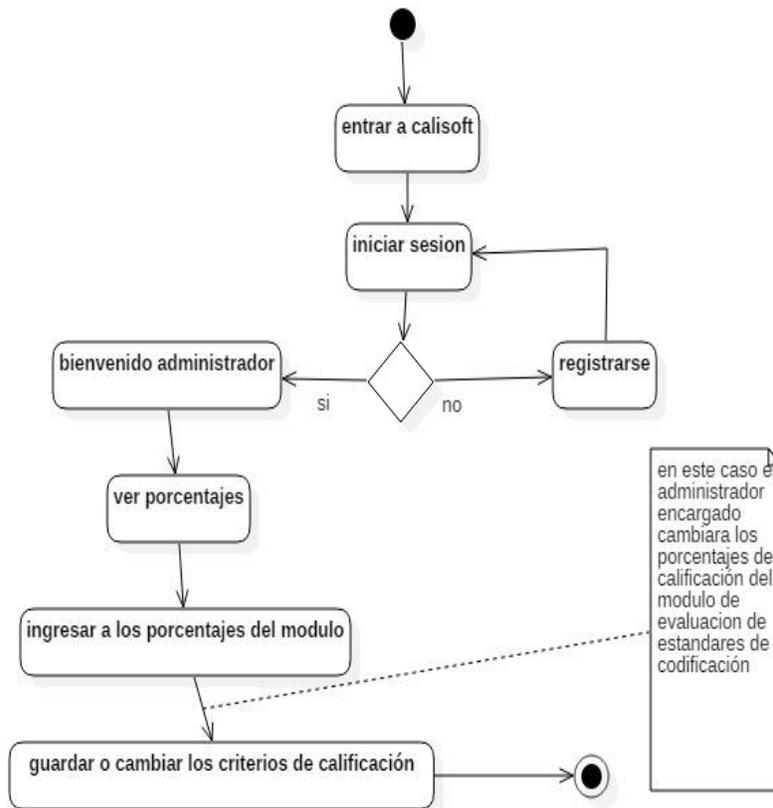


Figura 38. Diagrama de actividades cambiar criterios de evaluación.

Actividades	Descripción
Entrar a Calisoft	Inicio de la plataforma de Calisoft
Iniciar sesión	El usuario procede a acceder sus datos a el inicio de sesión
Bienvenido administrador	Si el administrador inicia sesión de manera exitosa procede a mostrar la página de inicio del administrador
Registrarse	El administrador inicia sesión de manera incorrecta, solicitar restaurar contraseña
Ver porcentajes	El administrador se dirige a ver los porcentajes
Ingresar a los porcentajes del modulo	El administrador ingresa a ver los porcentajes del módulo de evaluación de codificación
Guardar o cambiar los criterios de codificación	El administrador procede si tiene la necesidad de cambiar los criterios de evaluación del módulo de codificación
Ingresar al panel para la visualización del código fuente	Se muestra el código fuente del script seleccionado con los parámetros de calificación
Cerrar sesión	Una vez cumplida cualquiera de estas tareas el evaluador podrá cerrar sesión si lo desea

Tabla 33. Descripción diagrama de actividades cambiar criterio de evaluación.

5.5 Diagrama de clases

Es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, métodos u operaciones y las relaciones entre los objetos.

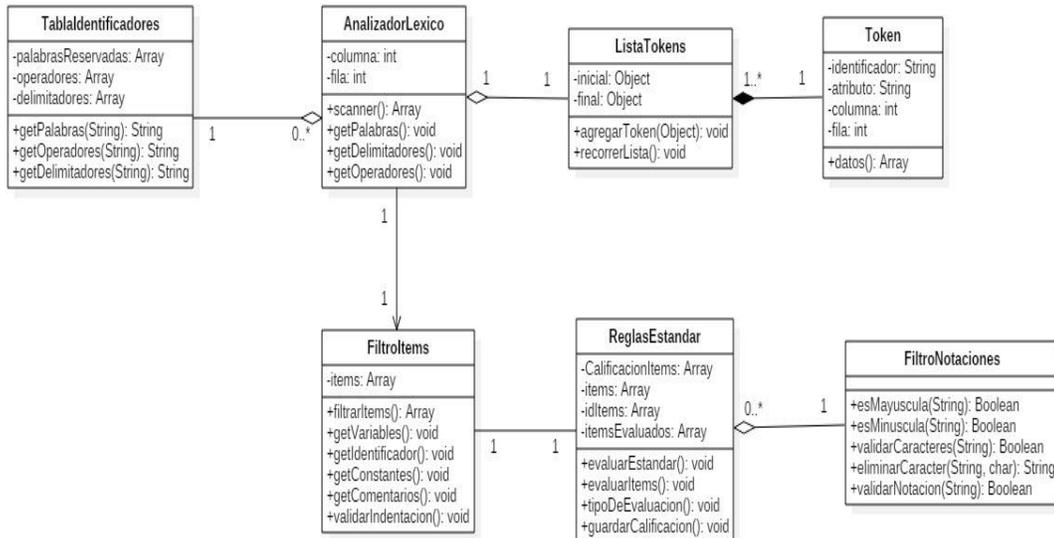


Figura 39. Diagrama de clases.

Clases	Descripción
AnalizadorLexico	Esta clase se encarga de leer el código fuente del archivo y de clasificar cada uno de estos según sea su tipo, ya sea una palabra reservada, un operador etc.
TablaIdentificadores	Esta clase almacena todas las palabras y operadores que hacen parte del léxico del lenguaje PHP. Es usada por la clase AnalizadorLexico como una clase de consulta para clasificar los elementos que se encuentran en el código fuente.
Token	Esta clase contiene información de un elemento del código fuente. Ejemplo: si es una palabra reservada almacena el valor de esta, su ubicación en el archivo y demás información para su posterior procesamiento.
ListaTokens	Esta clase es el producto del análisis léxico, esta es una estructura de datos que contiene elementos de la clase token enlazados entre sí, que pueden ser iterados para realizar
FiltroItems	Esta clase toma la lista de tokens y los filtra según el ítem de evaluación al cual se va a someter, luego los ordena según el ítem y
ReglasEstandar	Esta clase toma la lista de tokens filtrados y evalúa a cada uno de estos según las reglas del estándar que están previamente establecidas, también se encarga de calcular los promedios para cada uno de los ítems del estándar y por último almacena la calificación
FiltroNotaciones	Esta clase contiene una serie de funciones que permiten determinar si una cadena o string cumple con una notación específica, esta clase permite evaluar tres tipos de notación, lowerCamelCase, UpperCamelCase y snake_case, que son las

Tabla 34. Descripción diagrama de clases.

6. CASOS DE PRUEBA

6.1 Diseño casos de prueba

Caso de prueba	
Numero	001
Nombre	Caso prueba declaración de variables
Requerimientos	Fichero código fuente
Propósito	Verificar que el software evalué correctamente la notación establecida para la declaración de variables.
Actividades	La directriz de declaración de variables se evaluará de la siguiente forma, se cargara un archivo a la plataforma el cual contendrá variables declaradas con diferentes tipos de notaciones, el software deberá aprobar solo las variables que estén con la notación establecida por el estándar.
Resultados	El resultado de este caso de prueba será el reporte emitido por el software, el cual se analizara en la sección de resultados.

Tabla 35. Caso prueba declaración de variables.

Caso de prueba	
Numero	002
Nombre	Caso prueba declaración de funciones
Requerimientos	Fichero código fuente
Propósito	Verificar que el software evalué correctamente la notación establecida para la declaración de funciones.
Actividades	La evaluación de la directriz declaración de funciones se realizara cargando un fichero a la plataforma con que contendrá diferentes tipos notaciones, así el software deberá evaluar cada una de estas y determinar si cumplen con la notación establecida por el estándar.
Resultados	El resultado de este caso de prueba será el reporte emitido por el software, el cual se analizara en la sección de resultados.

Tabla 36. Caso prueba declaración de funciones.

Caso de prueba	
Numero	003
Nombre	Caso prueba declaración de clases
Requerimientos	Fichero código fuente
Propósito	Verificar que el software evalúe correctamente la notación establecida para la declaración de clases.
Actividades	La directriz de evaluación de declaración de clases se evaluara, cargando un fichero a la plataforma, con diferentes tipos de notación, el software deberá evaluar cada una de estas y verificar que correspondan con la notación establecida por el estándar.
Resultados	El resultado de este caso de prueba será el reporte emitido por el software, el cual se analizara en la sección de resultados.

Tabla 37. Caso prueba declaración de clases.

Caso de prueba	
Numero	004
Nombre	Caso prueba indentacion estructuras de control
Requerimientos	Fichero código fuente
Propósito	Verificar que el software evalúe correctamente la indentacion de las estructuras de control según el estándar previamente establecido.
Actividades	La directriz de evaluación de indentación de estructuras de control se evaluara, cargando un fichero de código fuente que contenga diferentes estructuras de control, las cuales estarán indentadas de una manera distinta el software deberá aprobar solo las estructuras que estén indentadas según el estándar.
Resultados	El resultado de este caso de prueba será el reporte emitido por el software, el cual se analizara en la sección de resultados.

Tabla 38. Caso prueba indentacion estructuras de control.

Caso de prueba	
Numero	005
Nombre	Caso prueba uso de comentarios en el código fuente
Requerimientos	Fichero código fuente
Propósito	Verificar que el software evalué correctamente el uso de comentarios en el código fuente.
Actividades	La directriz uso de comentarios en el código fuente se evaluara, cargando a la plataforma un fichero de código fuente el cual contendrá varias estructuras de código, las cuales según el estándar deben estar siempre acompañadas por su correspondiente comentario el software deberá aprobar solo las estructuras que cumplan con esta condición.
Resultados	El resultado de este caso de prueba será el reporte emitido por el software, el cual se analizara en la sección de resultados.

Tabla 39. Caso de prueba uso de comentarios en el código fuente.

Caso de prueba	
Numero	006
Nombre	Caso prueba declaración de constantes
Requerimientos	Fichero código fuente
Propósito	Verificar que el software evalué correctamente la notación establecida para la declaración de constantes.
Actividades	La directriz de evaluación de declaración de constantes se evaluará, cargando un fichero a la plataforma, con diferentes tipos de notación, el software deberá evaluar cada una de estas y verificar que correspondan con la notación establecida por el estándar.
Resultados	El resultado de este caso de prueba será el reporte emitido por el software, el cual se analizara en la sección de resultados.

Tabla 40. Caso prueba declaración de constantes.

Caso de prueba	
Numero	007
Nombre	Caso prueba declaración de espacios de nombre
Requerimientos	Fichero código fuente
Propósito	Verificar que el software evalué correctamente la notación establecida para la declaración de espacios de nombre.
Actividades	La directriz de evaluación de declaración de espacios de nombre se evaluara, cargando un fichero a la plataforma, con diferentes tipos de notación, el software deberá evaluar cada una de estas y verificar que correspondan con la notación establecida por el estándar.
Resultados	El resultado de este caso de prueba será el reporte emitido por el software, el cual se analizara en la sección de resultados.

Tabla 41. Caso prueba declaración de espacios de nombre.

7. ESTIMACION DE RECURSOS

7.1 Resumen por rubros

Rubros	Solicitado en efectivo a UDEC	Contrapartida en especie		Total
		UDEC	Otras Entidades	
PERSONAL	\$50'000.000			\$50'000.000
EQUIPOS	\$4'000.000			\$4'000.000
MATERIALES E INSUMOS	\$300.000			\$300.000
SERVICIOS TECNOLOGICOS	\$1'000.000			\$1'000.000
VIAJES	\$900.000			\$900.000
OTROS	\$2'000.000			\$2'000.000
TOTALES	\$56'700.000			\$58'200.000

Tabla 42. Resumen por rubros.

7.2 Detalle personal

Nombre	Función en el proyecto	Tipo de vinculación	Dedicación Horas/ semana	Entidad a la que pertenece	Solicitado en efectivo a UDEC	Total
Cesar Barahona	Investigador principal	Docente	ocho horas	UDEC	\$20'000.000	\$20'000.000
Héctor Castellanos	Estudiante pregrado investigador auxiliar	Estudiante	ocho horas	UDEC	\$15'000.000	\$15'000.000
John Osorio	Estudiante pregrado investigador auxiliar	Estudiante	ocho horas	UDEC	\$15'000.000	\$15'000.000

Tabla 43. Detalles personales.

7.3 Descripción de equipos

Descripción	Justificación	Cantidad	Valor Unitario	Solicitado en efectivo a UDEC	Total
Hewlett-Packard	Herramienta de trabajo	1	\$1'500.000	\$2'000.000	\$2'000.000
Lenovo	Herramienta de trabajo	1	\$1'500.000	\$2'000.000	\$2'000.000

Tabla 44. Descripción de equipos.

7.4 Descripción de materiales e insumos

Descripción	Justificación	Cantidad	Valor Unitario	Solicitado en efectivo a UDEC	Total
Papelería	Documentación	10	\$30.000	\$300.000	\$300.000

Tabla 45. Descripción de materiales e insumos.

7.5 Descripción de servicios tecnológicos

Descripción	Justificación	Valor	Entidad	Solicitado en efectivo a UDEC	Total
Conexión a internet	Fuente de Investigación	\$1'000.000	Claro	\$1'000.000	\$1'000.000

Tabla 46. Descripción de servicios tecnológicos.

7.6 Descripción de viajes

Lugar/justificación	No. días	No. personas	Cantidad	Valor Unitario	Solicitado en efectivo a UDEC	Total
Conferencias y Capacitaciones	3	2	3	\$150.000	\$900.000	\$900.000

Tabla 47. Descripción de viajes.

7.7 Descripción de otros

Descripción	Justificación	Cantidad	Valor Unitario	Solicitado en efectivo a UDEC	Total
Imprevistos	Daño de equipos	2	\$1.000.000	\$2.000.000	\$2'000.000

Tabla 48. Descripción de otros.

8. RESULTADOS

El software para la evaluación de estándares de codificación bajo la metodología de software de la universidad de Cundinamarca busca evaluar, si el código fuente de los proyectos desarrollados por los estudiantes, cumple con el estándar previamente establecido, para esto se definieron siete (7) casos de prueba que concuerdan con las normas establecidas por el estándar. Las normas que se tendrán en cuenta para la evaluación son declaración de variables, declaración de clases, declaración de funciones, indentación de estructuras de control, uso de comentarios en el código fuente, declaración de constantes y declaración de espacios de nombre, cada uno de estos aspectos se explica en el apartado del estándar. Para determinar si el código aprueba, se diseñó una métrica de evaluación, que asigna una prioridad a cada una de las directrices de evaluación, esta métrica arroja como resultado un valor porcentual el cual permite determinar la aprobación del proyecto.

Las pruebas del software se desarrollarán en base a cada una de los casos de prueba, observando como este evalúa los ficheros de código fuente y de que efectivamente apruebe solo los ítems que cumplen con las condiciones del estándar.

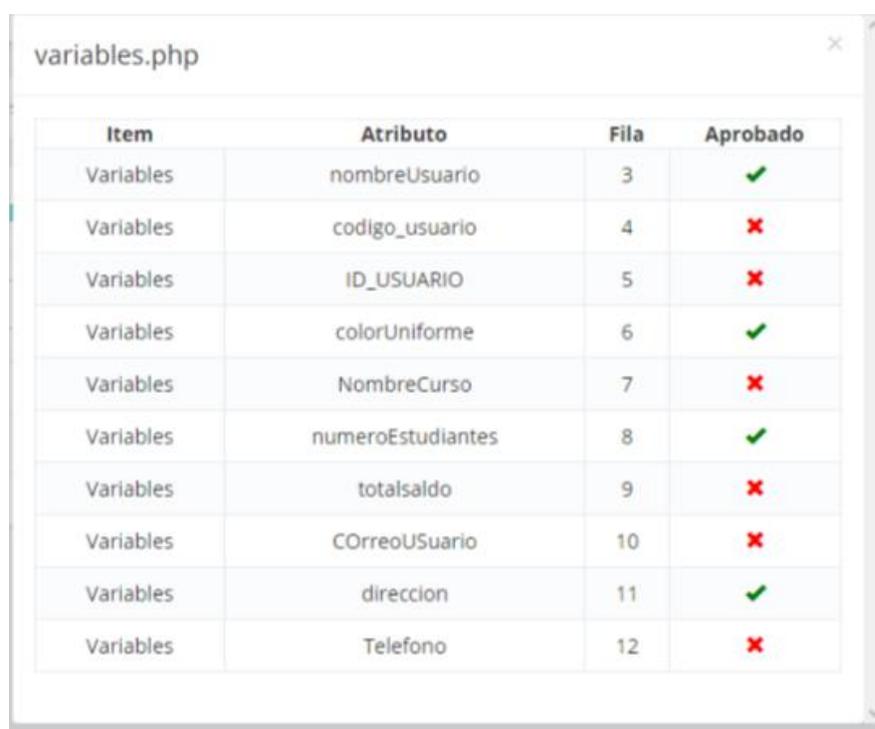
8.1 Evaluación declaración de variables

El proceso para realizar la evaluación de declaración de variables, se definió en el caso de prueba número 001 en el cual se definió la actividad a realizar para este caso, como requerimiento de entrada se dispone de un fichero de código fuente, en cual se utilizara para verificar el correcto funcionamiento del software.

EVALUAR DOCUMENTO: 8562_VARIABLES.PHP

```
1 <?php
2
3 $nombreUsuario = "carlos perez";
4 $codigo_usuario = 12345;
5 $ID_USUARIO = 676;
6 $colorUniforme = "Rojo";
7 $NombreCurso = "Sistemas";
8 $numeroEstudiantes = 30;
9 $totalsaldo = 600.89;
10 $COrrreoUSuario = "app@mail.com";
11 $direccion = "calle 1 b 30";
12 $Telefono = 8459023;
```

Figura 40. Fichero declaración de variables.



Item	Atributo	Fila	Aprobado
Variables	nombreUsuario	3	✓
Variables	codigo_usuario	4	✗
Variables	ID_USUARIO	5	✗
Variables	colorUniforme	6	✓
Variables	NombreCurso	7	✗
Variables	numeroEstudiantes	8	✓
Variables	totalsaldo	9	✗
Variables	COrrreoUSuario	10	✗
Variables	direccion	11	✓
Variables	Telefono	12	✗

Figura 41. Reporte evaluación fichero variables.php.

En la figura 40. Se muestra un fichero de código fuente el cual contiene la declaración de diez (10) variables, en la figura 41. Se presenta el reporte de evaluación de ese fichero en cual presenta el ítem de evaluación, en este caso variables, el nombre de la variable, la ubicación dentro del fichero y si el ítem fue aprobado o no, se puede ver que solo los que cumplen con la notación

lowerCamelCase son los que se aprobaron, los demás se toman como erróneos.

8.2 Evaluación declaración de funciones

El proceso para realizar la evaluación de declaración de funciones, se definió en el caso de prueba número 002 en el cual se definió la actividad a realizar para este caso, como requerimiento de entrada se dispone de un fichero de código fuente, el cual se utilizara para verificar el correcto funcionamiento del software.

EVALUAR DOCUMENTO: 1052_FUNCIONES.PHP

```
1 <?php
2
3 function calcularSaldo(){ return true; }
4 function REGISTRAR_SALDO(){ return false; }
5 function ValidarNombre(){ return true; }
6 function sumarCuentas(){ return false; }
7 function ResTaRAcumlados(){ return true; }
8 function registrar_cuenta(){ return true; }
9 function listarProductosNuevos(){ return false; }
10 function cerrarsesion(){ return true; }
11 function CAMBIARClave(){ return false; }
12 function seleccionaREpleado(){ return true; }
13
```

Figura 42. Fichero declaración de funciones.

Item	Atributo	Fila	Aprobado
Funciones	calcularSaldo	3	✓
Funciones	REGISTRAR_SALDO	4	✗
Funciones	ValidarNombre	5	✗
Funciones	sumarCuentas	6	✓
Funciones	ResTaRAcumlados	7	✗
Funciones	registrar_cuenta	8	✗
Funciones	listarProductosNuevos	9	✓
Funciones	cerrarsesion	10	✗
Funciones	CAMBIARClave	11	✗
Funciones	seleccionaREmpleado	12	✗

Figura 43. Reporte evaluación fichero funciones.php.

En la figura 42. Se observa el fichero funciones que contiene la declaración de diez (10) funciones, en la figura 43. Se evidencia el reporte generado por el software en el cual se ve que solo los nombres de funciones que cumplen con la notación lowerCamelCase se encuentran aprobados.

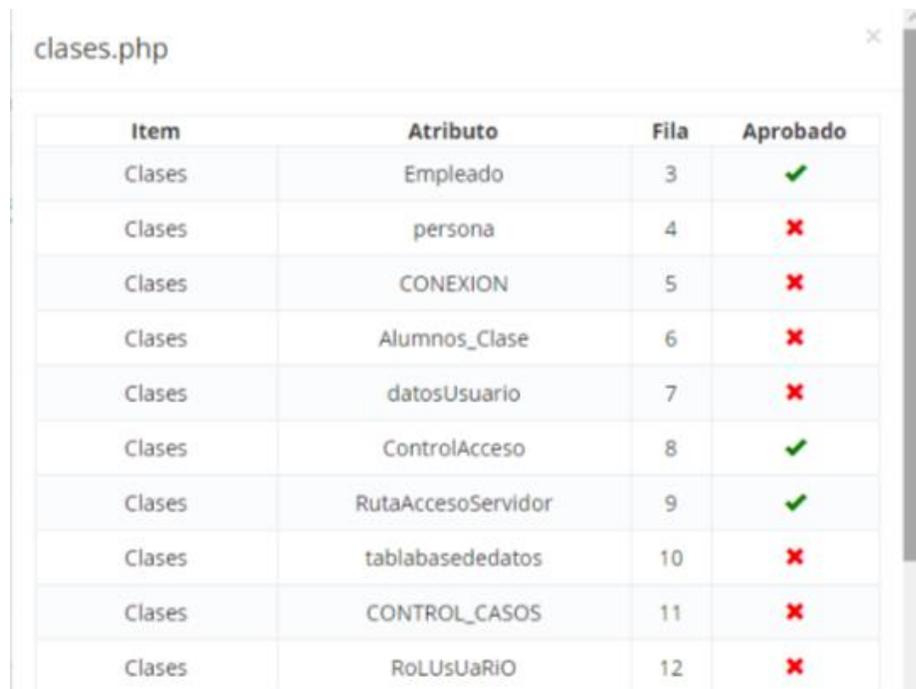
8.3 Evaluación declaración de clases

El proceso para realizar la evaluación de declaración de clases, se definió en el caso de prueba número 003 en el cual se definió la actividad a realizar para este caso, como requerimiento de entrada se dispone de un fichero de código fuente, el cual se utilizara para verificar el correcto funcionamiento del software.

EVALUAR DOCUMENTO: 6275_CLASES.PHP

```
1 <?php
2
3 class Empleado { }
4 class persona { }
5 class CONEXION { }
6 class Alumnos_Clase { }
7 class datosUsuario { }
8 class ControlAcceso { }
9 class RutaAccesoServidor { }
10 class tablabasededatos { }
11 class CONTROL_CASOS { }
12 class RoLUuUaRiO { }
```

Figura 44. Fichero declaración de clases.



Item	Atributo	Fila	Aprobado
Clases	Empleado	3	✓
Clases	persona	4	✗
Clases	CONEXION	5	✗
Clases	Alumnos_Clase	6	✗
Clases	datosUsuario	7	✗
Clases	ControlAcceso	8	✓
Clases	RutaAccesoServidor	9	✓
Clases	tablabasededatos	10	✗
Clases	CONTROL_CASOS	11	✗
Clases	RoLUuUaRiO	12	✗

Figura 45. Reporte evaluación fichero clases.php.

En la figura 44. Se presenta el fichero clases que contiene la declaración de diez (10) clases cada una con una notación diferente, en la figura 45. Se observa el reporte de evaluación del fichero, en cual se evidencia que solo las clases que están con la notación UpperCamelCase se encuentran aprobadas por el software.

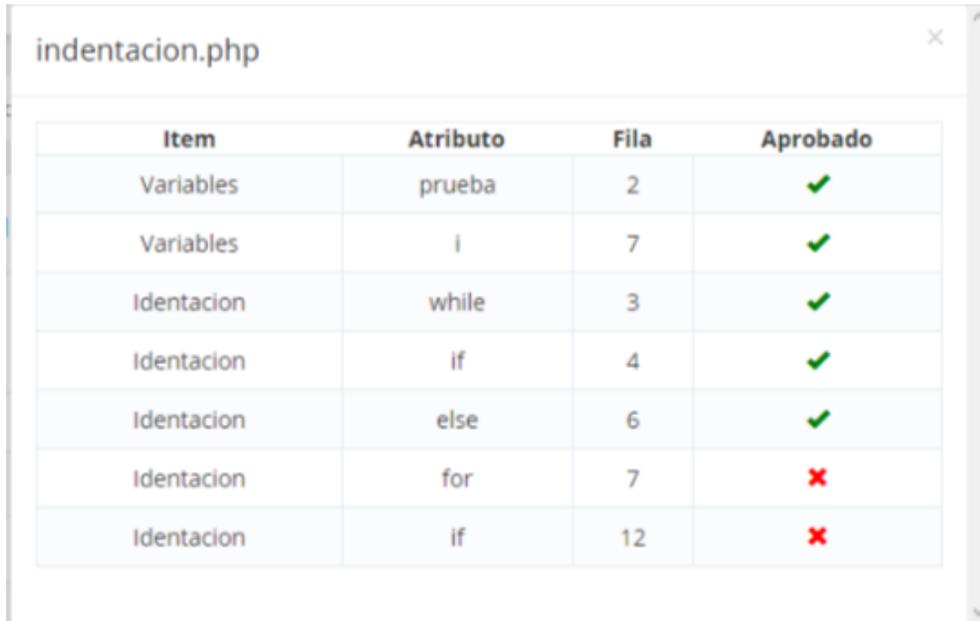
8.4 Evaluación indentacion estructuras de control

El proceso para realizar la evaluación de indentacion de estructuras de control, se definió en el caso de prueba número 004 en el cual se definió la actividad a realizar para este caso, como requerimiento de entrada se dispone de un fichero de código fuente, el cual se utilizara para verificar el correcto funcionamiento del software.

EVALUAR DOCUMENTO: 7652_INDENTACION.PHP

```
1 <?php
2 $prueba = 10;
3 while(true){
4     if($prueba == 10){
5         echo $prueba;
6     }else{
7         for($i = 0; i<10; $i++){
8             echo $i;
9             $prueba--;
10        }
11    }
12    if($prueba == 0){
13        echo "Este es un fichero prueba";
14        $prueba = 10;
15    }
16 }
```

Figura 46. Fichero estructuras de control.



Item	Atributo	Fila	Aprobado
Variables	prueba	2	✓
Variables	i	7	✓
Indentacion	while	3	✓
Indentacion	if	4	✓
Indentacion	else	6	✓
Indentacion	for	7	✗
Indentacion	if	12	✗

Figura 47. Reporte de evaluación fichero indentacion.php.

En la figura 46. Se presenta el fichero indentación que contiene varias estructuras de control como ciclos y condicionales, en la figura 47. Se observa el reporte de evaluación del fichero, en este se puede evidenciar que solo las estructuras que están correctamente indentadas fueron aprobadas por el software.

8.5 Evaluación uso de comentarios en el código fuente

El proceso para realizar la evaluación de uso de comentarios en el código fuente, se definió en el caso de prueba número 005 en el cual se definió la actividad a realizar para este caso, como requerimiento de entrada se dispone de un fichero de código fuente, el cual se utilizara para verificar el correcto funcionamiento del software.

EVALUAR DOCUMENTO: 5945_COMENTARIOS.PHP

```
1 <?php
2 /* Declaracion clase usuario */
3 class Usuario {
4
5     function getNombreUsuario(){
6         return "Usuario";
7     }
8     // retorna edad del usuario
9     function getEdad(){
10        return 20;
11    }
12
13    function totalSaldo(){
14        return 200.45;
15    }
16 }
```

Figura 48. Fichero uso de comentarios en el código fuente.

Item	Atributo	Fila	Aprobado
Clases	Usuario	3	✓
Funciones	getNombreUsuario	5	✓
Funciones	getEdad	9	✓
Funciones	totalSaldo	13	✓
Identacion	function	5	✓
Identacion	function	9	✓
Identacion	function	13	✓
Comentarios	class	3	✓
Comentarios	function	5	✗
Comentarios	function	9	✓
Comentarios	function	13	✗

Figura 49. Reporte de evaluación fichero comentarios.php.

En la figura 48. Se presenta el fichero comentarios en el cual se encuentran varias estructuras de código, en la Figura 49. Se observa el reporte de evaluación del fichero en el cual se muestra que las funciones declaradas que no están acompañadas en su inicio por un comentario no fueron aprobadas por el software.

8.6 Evaluación declaración de constantes

El proceso para realizar la evaluación de declaración de constantes, se definió en el caso de prueba número 006 en el cual se definió la actividad a realizar para este caso, como requerimiento de entrada se dispone de un fichero de código fuente, el cual se utilizara para verificar el correcto funcionamiento del software.

EVALUAR DOCUMENTO: 9994_CONSTANTES.PHP

```
1 <?php
2
3 define("NOMBRE_USUARIO","admin");
4 define("fecha_nacimiento","10/3/1979");
5 define("Grupo_Sanguineo","0");
6 define("URL","https://es.wikipedia.org");
7 define("cuentaSaldo",542365);
8 define("LONGITUD_DE_COMPONENTE",200.45);
9 define("ModoDeCierre",2);
10 define("UsuArio_RegistRad0",true);
11 define("ANCHO_CELDA",27);
12 define("Estatura",1.87);
13
```

Figura 50. Fichero declaración de constantes.



Item	Atributo	Fila	Aprobado
Constantes	NOMBRE_USUARIO	3	✓
Constantes	fecha_nacimiento	4	✗
Constantes	Grupo_Sanguineo	5	✗
Constantes	URL	6	✓
Constantes	cuentaSaldo	7	✗
Constantes	LONGITUD_DE_COMPONENTE	8	✓
Constantes	ModoDeCierre	9	✗
Constantes	UsuArio_RegistRad0	10	✗
Constantes	ANCHO_CELDA	11	✓
Constantes	Estatura	12	✗

Figura 51. Reporte evaluación fichero constantes.php.

En la figura 50. Se presenta el fichero constantes que contiene la declaración de diez (10) constantes cada una con una notación diferente, en la figura 51. Se observa el reporte de evaluación del fichero, en cual se evidencia que solo

las constantes que están con la notación Snake case se encuentran aprobadas por el software.

8.7 Evaluación declaración de espacios de nombre

El proceso para realizar la evaluación de declaración de espacios de nombre, se definió en el caso de prueba número 007 en el cual se definió la actividad a realizar para este caso, como requerimiento de entrada se dispone de un fichero de código fuente, el cual se utilizara para verificar el correcto funcionamiento del software.

EVALUAR DOCUMENTO: 5705_ESPACIOS DE NOMBRE.PHP

```
1 <?php
2 namespace Modelos {
3     // Declaraciones
4 }
5 namespace CONTROLADORES {
6     // Declaraciones
7 }
8 namespace vistas {
9     // Declaraciones
10 }
11 namespace Rutas_de_acceso {
12     // Declaraciones
13 }
14 namespace FiltrosControles {
15     // Declaraciones
16 }
```

Figura 52. Fichero declaración espacios de nombre.



Item	Atributo	Fila	Aprobado
Espacios De Nombre	Modelos	2	✓
Espacios De Nombre	CONTROLADORES	5	✗
Espacios De Nombre	vistas	8	✗
Espacios De Nombre	Rutas_de_acceso	11	✗
Espacios De Nombre	FiltrosControles	14	✓

Figura 53. Reporte de evaluación fichero espacios de nombre.php

En la figura 52. Se presenta el fichero espacios de nombre que contiene la declaración de cinco (5) espacios de nombre cada una con una notación diferente, en la figura 53. Se observa el reporte de evaluación del fichero, en cual se evidencia que solo las constantes que están con la notación UpperCamelCase se encuentran aprobadas por el software.

9. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- En el desarrollo de este proyecto se pudo establecer un estándar que estuviera acorde a las tecnologías utilizadas actualmente, dando a entender que el uso de estas prácticas aún está vigente y es altamente utilizado.
- Se pudo diseñar una métrica de evaluación, que busca dar una forma diferente de medir el nivel de calidad de un código fuente.
- Se puede desarrollar software que automatice este tipo de procesos como es la revisión de código fuente, para agilizar las evaluaciones de calidad en los productos informáticos.

Recomendaciones

- En el momento de evaluar la notación de los diferentes elementos del código, el software no siempre evalúa de la forma correcta, puede presentar fallos en este aspecto, se recomienda en un futuro desarrollo optimizar este proceso para lograr minimizar el error y mejorar el funcionamiento del software.
- En este proyecto se trabajó sobre un estándar del lenguaje PHP, se recomienda que si se quieren agregar nuevas funcionalidades, se puede abarcar estándares para otras tecnologías como JavaScript, HTML y CSS, y realizar un proceso similar al presentado en este trabajo.

REFERENCIAS

- Ala-Mutka, K., Uimonen, T., & Jarvinen, H.-M. (2004). Supporting Students in C++ Programming Courses with Automatic Program Style Assessment. *Journal of Information Technology Education*, 3(1), 245–262. Retrieved from http://search.proquest.com/docview/61852712?accountid=14570%5Cnhtp://sfx.bib.uni-mannheim.de:8080/sfx_local?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&genre=article&sid=ProQ:ProQ:ericshell&atitle=Supporting+Students+in+C++Programming+C
- Boogerd, C., & Moonen, L. (2008). Assessing the value of coding standards: An empirical study. In *2008 IEEE International Conference on Software Maintenance* (pp. 277–286). IEEE. <https://doi.org/10.1109/ICSM.2008.4658076>
- Boogerd, C., & Moonen, L. (2009a). Evaluating the relation between coding standard violations and faults within and across software versions. *Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories, MSR 2009*, 41–50. <https://doi.org/10.1109/MSR.2009.5069479>
- Boogerd, C., & Moonen, L. (2009b). Using software history to guide deployment of coding standards. *Trader: Reliability of High-Volume Consumer Products*, 39–52.
- Dai, Z., Mao, X., Wang, D., Liu, D., & Zhang, J. (2010). Checking compliance to coding standards for x86 executables. *Proceedings - Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing in Conjunction with the UIC 2010 and ATC 2010 Conferences, UIC-ATC 2010*, (November 2010), 449–455. <https://doi.org/10.1109/UIC-ATC.2010.118>
- Fernández Romero Yenisleidy, D. G. Y. (2012). Patrón Modelo-Vista-Controlador. *Revista Telem@tica*, 11(1), 11. Retrieved from <http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/10>
- Hunt, L. (2012). Coding Standards for .NET, (March), 1–33. Retrieved from <http://se.inf.ethz.ch/old/teaching/ss2007/251-0290-00/project/CSharpCodingStandards.pdf>
- Java Coding Conventions. (1997). Retrieved from

<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

Louise, R. D. (2009). If You Have Programming Standards , Please Raise Your Hand : An Everyman ' s Guide, 1–8. Retrieved from <http://analytics.ncsu.edu/sesug/2010/PS01.Rhodes.pdf>

MacKellar, D. V. (2016). Injection of Business Coding Standards Practices to Embedded Software Courses Donald. In *2016 IEEE Frontiers in Education Conference (FIE)* (pp. 1–8). IEEE. <https://doi.org/10.1109/FIE.2016.7757469>

Michaelson, G. (2015). Automatic analysis of functional program style. *Proceedings of 1996 Australian Software Engineering Conference*, (January), 38–46. <https://doi.org/10.1109/ASWEC.1996.534121>

MISRA. (2016). MISRA Compliance:2016 Achieving compliance with MISRA Coding Guidelines. Retrieved from https://www.misra.org.uk/LinkClick.aspx?fileticket=w_Syhpkf7xA%3D&tabid=57

Pulido, S., & Rodríguez, J. (2014). Manual básico de uso: Software NVivo. V 9 &10. Retrieved from [http://www.fce.unal.edu.co/uifce/proyectos-de-estudio/pdf/Manual Nvivo 10](http://www.fce.unal.edu.co/uifce/proyectos-de-estudio/pdf/Manual%20Nvivo%2010)

Roa, P., Morales, C., & Gutierrez, P. (2015). Norma ISO / IEC 25000. *Universidad Distrital Francisco Jose De Caldas*, 3(2).

Rodríguez, J. C., Pino, D., Roca, M. D., Figueroa, Z. H., & González Domínguez, D. (n.d.). Hacia la Evaluación Continua Automática de Prácticas de Programación. Retrieved from https://www.researchgate.net/profile/Juan_Rodriguez-Del-Pino/publication/237228089_Hacia_la_Evaluacion_Continua_Automatica_de_Practicas_de_Programacion/links/02e7e53c67cc8b980d000000/Hacia-la-Evaluacion-Continua-Automatica-de-Practicas-de-Programacion.pdf

Universidad de Sonora. (n.d.). La metodología cuantitativa. Encuestas y muestras. Retrieved from <http://www.um.es/docencia/pguardio/documentos/master2.pdf>

ANEXOS

Anexo A. Manual de usuario



UDEC
UNIVERSIDAD DE
CUNDINAMARCA

CALISOFT UDEC



DESARROLLO DE SOFTWARE UNIVERSIDAD DE CUNDINAMARCA

MANUAL DE USUARIO 2018

www.unicundi.edu.co
unicundi@mail.unicundi.edu.co
Línea gratuita 018000 976000



Dirección de Sistemas y Tecnología
sistemasytecnologia@mail.unicundi.edu.co
PBX: 828 14 83 Ext. 110-170
Sede Fusagasugá

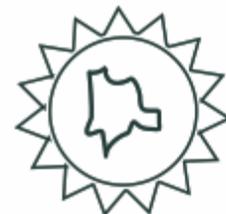
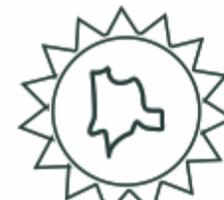


TABLA DE CONTENIDO

INTRODUCCIÓN.....	127
1. USUARIOS.....	128
1.1. ADMINISTRADOR.....	128
1.2. EVALUADOR.....	128
1.3. ESTUDIANTE.....	129
2. MÓDULOS.....	130
2.1. MODULOS BÁSICOS.....	130
2.2. MÓDULO DE MODELADO.....	130
2.3. MÓDULO DE TESTING.....	131
2.4. MÓDULO DE CODIFICACIÓN.....	131
2.5. MÓDULO DE BASES DE DATOS.....	132
3. REQUISITOS DE SOFTWARE Y HARDWARE.....	133
3.1. REQUERIMIENTOS MÍNIMOS DE HARDWARE.....	133
3.2. REQUERIMIENTOS MÍNIMOS DE SOFTWARE.....	133
4. APLICATIVO CALISOFT.....	134
4.1. ICONOS GENERALES.....	134
4.2. INGRESO AL APLICATIVO.....	134
4.2.1. Página principal.....	135
4.2.2. Inicio de sesión.....	136
4.2.3. Registro.....	137
4.3. USO DEL APLICATIVO.....	138



4.3.1. Administrador	138
4.3.2. Evaluador	139
4.3.3. Desarrollador.....	150
5. CONTROL DE CAMBIOS DEL MANUAL	163

TABLA DE IMÁGENES

Imagen 1 Página principal	135
Imagen 2 Componentes página principal.....	135
Imagen 3 Inicio sesión	136
Imagen 4 Enlace recuperar contraseña	137
Imagen 5 Vista recuperar contraseña.....	137
Imagen 6 Enlace de registro	137
Imagen 7 Vista de registro de usuario	138
Imagen 8 Porcentajes de codificación	139
Imagen 9 Dashboard evaluador.....	140
Imagen 10 Porcentajes de codificación evaluador.....	141
Imagen 11 Documentación de los estándares de codificación	141
Imagen 12 Tabla de Archivos de codificación del evaluador	142
Imagen 13 Modal ayuda de lista de Scripts	142
Imagen 14 Panel de calificación de Scripts	143
Imagen 15 Modal de ayuda para Calificar un Script	144
Imagen 16 Ejemplo Block UI.....	145
Imagen 17 Panel de Calificación Manual.....	145
Imagen 18 Panel de Calificación Manual con Resultados	146
Imagen 19 Panel de comentarios	146
Imagen 20 Diferencia entre Script calificado y sin calificar	147
Imagen 21 Tabla de Ítems sin Evaluar Evaluador.....	147
Imagen 22 Tabla de ítems Evaluados.....	148
Imagen 23 Reporte del módulo para la evaluación de codificación Evaluador	149
Imagen 24 Nota final calculando el índice de cohesión o aceptación evaluador .	149
Imagen 25 Proceso final de calificación de proyectos	150
Imagen 26 Dashboard del desarrollador.....	151

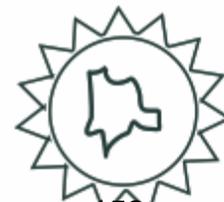


Imagen 27 Registro del proyecto	152
Imagen 28 Proyecto listo para enviarlo como propuesta	153
Imagen 29 Modal invitar usuario	153
Imagen 30 Usuarios invitados.....	154
Imagen 31 Aceptar invitación.....	154
Imagen 32 Sección de integrantes actualizada	155
Imagen 33 Proyecto listo para evaluación	156
Imagen 34 Primera Vista del módulo de codificación	156
Imagen 35 Modal de ayuda para subir archivos de codificación.....	157
Imagen 36 Panel de carga para subir Archivos de codificación.....	157
Imagen 37 Panel de Carga con Archivos de codificación	158
Imagen 38 tabla de listado de archivos de codificación estudiante.....	158
Imagen 39 Modal para eliminar un Script	159
Imagen 40 Panel de Visualización del código del desarrollador	159
Imagen 41 Proyecto en estado de evaluación	160
Imagen 42 Panel de Archivos de codificación calificados en la vista del desarrollador	161
Imagen 43 Tabla de Ítems sin Evaluar en la vista del desarrollador	161
Imagen 44 Tabla de Ítems evaluados en la vista del desarrollador	162
Imagen 45 Reporte del módulo para la evaluación de codificación Desarrollador	163
Imagen 46 Nota final calculando el índice de cohesión o aceptación Desarrollador	163

TABLA DE TABLAS

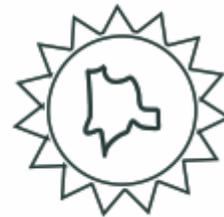
Tabla 1 Iconos generales.....	134
-------------------------------	-----



INTRODUCCIÓN

El aplicativo Calisoft es una plataforma web destinada a brindar soporte a la calificación de productos software que se presenten dentro de la Universidad de Cundinamarca, con el fin de realizar una evaluación estandarizada que certifica la calidad del producto a evaluar.

El proceso consiste en que el desarrollador debe subir a la plataforma segmentos de su producto software siguiendo un orden que la misma estipula, posteriormente los segmentos serán evaluados por un evaluador capacitado en la usabilidad del software, y este procederá a realizar las diferentes pruebas en los diferentes segmentos, para así generar un reporte final que certifique en calidad al producto software evaluado.



1. USUARIOS

1.1. ADMINISTRADOR

El administrador es el usuario encargado de administrar y estipular los estándares de calificación de los diferentes componentes de la plataforma. Además de dar roles y aceptar los usuarios que se registren en la plataforma. Entre sus funciones principales encontramos las siguientes:

- Registra semilleros y grupos de investigación que estén vigentes en la Universidad de Cundinamarca, con el objetivo de agrupar los proyectos que se presenten.
- Administrar usuarios, esto le permite al administrador modificar los datos de los usuarios que se encuentren registrados en la base de datos, al igual que crear, deshabilitar y aceptar o denegar peticiones.
- Administración de la documentación que se le exigirá al desarrollador, y además define los porcentajes de evaluación de cada documento, esto es necesario al momento de realizar la evaluación final.
- Administra las categorías en donde se irán agrupando los productos software a evaluar, para así llevar un orden en su estructura y almacenamiento.
- Administración de proyectos, en el momento en que un desarrollador cree un proyecto, este será enviado al administrador, el cual decidirá si aceptar o no dicho proyecto, una vez aceptado, el administrador deberá asignarle un evaluador que tendrá como objetivo realizar todo el proceso de calificación al proyecto.

1.2. EVALUADOR

El evaluador es el usuario encargado de realizar todo el proceso de calificación de calidad de los proyectos que tenga asignado, para realizar una correcta calificación, el evaluador debe estudiar y analizar a fondo todos los documentos proporcionados por el desarrollador, posteriormente procederá a hacer uso de los diferentes módulos de Calisoft para realizar las correspondientes pruebas. Entre sus funciones principales encontramos las siguientes:

- Interactuar con los proyectos asignados le permite al evaluador ver la documentación de todo el proyecto que suba el desarrollador.
- Calificar diagramas y modelado del aplicativo.
- Evaluar el código fuente del producto software mediante el módulo de codificación.
- Evaluar la nomenclatura de la base de datos por medio del módulo de base de datos



- Crear casos de prueba correspondientes a los casos de uso que el desarrollado haya estipulado en su documentación.
- Por medio de los casos de prueba el desarrollador procederá a anexar una copia de los atributos de los campos de entrada que pertenezcan a dicho caso de prueba, con la copia de los atributos se procede a realizar un escenario de pruebas, Calisoft procede a recrear los mismos campos de entrada gracias a la copia de los atributos obtenida por el desarrollador, y se procede a realizar pruebas no heurísticas automatizadas a los diferentes campos, con el objetivo de verificar la calidad en la seguridad al momento de ingresar datos erróneos o maliciosos al sistema, este procedimiento se realiza mediante el módulo de plataforma.
- Generar reportes finales, los cuales llevan una calificación que se obtiene mediante la suma y el promedio de los datos obtenidos de todos los módulos.
- Agregar observaciones sobre inconformidades o sugerencias sobre los productos software que tenga asignado.

1.3. ESTUDIANTE

El usuario tipo estudiante es asignado a los desarrolladores que pertenezcan a un proyecto, los desarrolladores deberán ingresar a la página principal del aplicativo y registrarse para acceder a sistema, dentro del sistema el desarrollador tendrá como primer objetivo, el de crear su proyecto y agregar a los demás autores si es necesario. Una vez creado el proyecto, el desarrollador podrá realizar las siguientes funciones dentro del sistema:

- Enviar el proyecto como propuesta al administrador, el administrador aceptará o rechazará el proyecto, si el proyecto es aceptado, se le asignará los evaluadores al proyecto.
- Subir toda la documentación referente al proyecto, tanto diagramas como estructura del código y de base de datos.
- Consultar las métricas de evaluación de los diferentes módulos del aplicativo Calisoft.
- Habilitar el proyecto para que inicie el proceso de evaluación. Una vez habilitado no se podrá realizar modificaciones a la documentación.
- Una vez terminado el proceso de evaluación del proyecto, se le presentará al desarrollador los diferentes resultados por medio de reportes agrupados en; pruebas de Testing, Evaluación de modelado, evaluación de estándares de codificación y evaluación de nomenclatura de bases de datos. Estos resultados si son negativos, los acompañara un apartado de observaciones, donde se le indica al desarrollador los cambios que debe realizar y su proyecto puede ser habilitado para que realice las correspondientes modificaciones.



2. MÓDULOS

2.1. MÓDULOS BÁSICOS

La plataforma está compuesta por cuatro módulos generales que tienen como objetivo realizar pruebas para evaluar la calidad y usabilidad en los productos software que se le presenten, estos módulos son; el módulo de modelado, el módulo de Testing, el módulo de calificación y el módulo de bases de datos. Lo que se pretende en este apartado es hablar sobre los módulos pequeños y básicos que no son menos importantes y que hacen en su conjunto a la plataforma Calisoft.

Dentro de los módulos básicos podremos cambiar la información personal que tengamos ligada a nuestra cuenta, además de modificar la imagen de perfil y cambiar la contraseña. También se cuenta con el módulo de notificaciones, el cuál tiene como objetivo tener una información histórica del movimiento de nuestra cuenta, proyectos y las interacciones que tienen los demás usuarios con nuestro perfil. Y por último tenemos el módulo de manuales y guías, donde los usuarios podrán observar como es el correcto funcionamiento de la plataforma y tengan un soporte para realizar las operaciones de la forma correcta.

2.2. MÓDULO DE MODELADO

El módulo de modelado es abarcado por los tres tipos de usuarios que tiene el sistema, el administrador, como usuario principal, deberá crear los respectivos documentos que se le exigirá al desarrollador, por defecto el aplicativo crea seis documentos, que son; diagrama de clases, casos de uso, secuencia, actividades, despliegue y entidad relación. Estos documentos ya vienen con sus respectivos componentes, el administrador es libre de modificarlos bajo su criterio o de crear nuevos documentos. Por el lado del desarrollador, debe crear su proyecto y enviarlo, una vez aceptado se le habilitara el módulo de modelación y por consiguiente la opción de subir la documentación correspondiente según el tipo de documento. Una vez cargados todos los documentos, el desarrollador debe cambiar el estado de su proyecto a evaluación, hecho esto no se podrá modificar ningún documento. En este punto del proceso al evaluador se le deberá habilitar la opción de calificar el proyecto, en la sección de modelado podrá ver todos los documentos que subió el estudiante y podrá continuar a realizar la evaluación, deberá corroborar estén todos los componentes de los documentos y que su uso sea el correcto en la estructura del diagrama. Al finalizar la evaluación, la calificación será dada por el número de componentes que tenga empleados correctamente.



2.3. MÓDULO DE TESTING

El módulo de Testing es el encargado de gestionar la seguridad de los métodos de entrada que tengan los productos software a evaluar, su funcionamiento se basa en la creación de casos de prueba que van acorde con los casos de uso del proyecto que sube el desarrollador a la plataforma Calisoft, el evaluador es el encargado de crear dichos casos de prueba, una vez creado el caso prueba, se le notificará al desarrollador, el desarrollador deberá descargar e instalar la extensión que se aloja en la página principal. Por medio de la extensión, el desarrollador deberá dirigirse a la página de su aplicativo y dirigirse al formulario del que pertenece el caso prueba que creo que evaluador, al activar la extensión en la página donde se encuentra el formulario, la extensión guardará todos los atributos de los inputs del formulario en un texto tipo Json, el desarrollador deberá dirigirse al módulo de Testing (plataforma) seleccionar el caso prueba que corresponda y en un campo tipo Text-Área deberá pegar el texto tipo Json y de esta forma el desarrollador habrá compartido su formulario con el evaluador. Al evaluador le llegará una notificación del caso prueba actualizado, y este procederá a ejecutar las pruebas automáticas no heurísticas sobre la copia del formulario que subió el desarrollador, corroborando así, que la forma en que el desarrollador tiene creado su formulario no permita el ingreso de datos erróneos o maliciosos, dependiendo de la cantidad de errores se le dará un reporte junto con una calificación que definirá si pasa o no la prueba de calidad. En caso de perderla, el desarrollador tiene la oportunidad de realizar los cambios que se adjunten en el reporte y volver a enviar su formulario.

2.4. MÓDULO DE CODIFICACIÓN

El módulo de evaluación de estándares de codificación es el encargado de verificar que los ficheros de código fuente desarrollados, cumplan con el estándar de codificación previamente establecido. Sobre el módulo trabajan tres usuarios anteriormente especificados, cada uno de estos cumplen funciones determinadas dentro del módulo, el administrador es el encargado de aceptar las solicitudes de los proyectos, también se encarga de asignar los evaluadores a los respectivos proyectos y puede cambiar los criterios de evaluación del módulo. El segundo usuario de este módulo es el estudiante, este se encarga de registrar su proyecto, y a su vez de cargar a la plataforma todos los ficheros de código fuente si el proyecto se encuentra habilitado este podrá modificar y eliminar sus ficheros. El tercer usuario de este módulo es el evaluador este se encarga de realizar la revisión de que los ficheros de código fuente cumplan con el estándar establecido, para realizar este proceso el evaluador, lo puede hacer de dos formas, una automática y una manual, en la forma manual el evaluador deberá registrar en cada una de las directrices el número total de ítems encontrados y el número de ítems aprobados, el proceso de calificación lo hará la plataforma, el podrá agregar un comentario sobre cada evaluación de fichero, la otra forma de hacerlo es la automática en esta el evaluador solo deberá seleccionar el fichero y dar click al botón de evaluación el



cual se encargara de todo el proceso de evaluación. Por último el evaluador y el estudiante podrán ver el resultado de la evaluación a través de un reporte generado por el sistema, el cual contendrá un resumen de cada uno de los archivos que componen el proyecto, este informe también posee el campo en el cual se indicara el índice de cohesión o aceptación del proyecto.

2.5. MÓDULO DE BASES DE DATOS

El módulo de base de datos es el encargado de hacer cumplir los estándares de calidad en la nomenclatura en el código SQL, permitiendo esto, tener un desarrollo de software adecuado. Su funcionamiento se basa en calificación automática y manual del código SQL, esto se desarrolla en el módulo del evaluador, el cual tiene la opción de hacer un análisis del archivo SQL automáticamente con ayuda del software, o que el evaluador haga la calificación el mismo, llamándose esta la calificación automática. El estudiante cumple el rol de subir el archivo SQL que después será calificado por el evaluador y también tendrá una interfaz gráfica para poder diseñar y modelar su base de datos desde el software de Calisoft, permitiéndole diseñar y modelar una base de datos desde el propio software y que desde esta misma interfaz puedan exportar el código SQL que les genera Calisoft desde dicha interfaz gráfica y posteriormente hacer uso de este código para que sea calificado por el evaluador. El módulo solo da un porcentaje de la calificación general, teniendo en cuenta los otros tres módulos, por lo que tener una calificación alta en el módulo de base de datos, puede ayudar a nivelar la calificación con los demás módulos.



3. REQUISITOS DE SOFTWARE Y HARDWARE

3.1. REQUERIMIENTOS MÍNIMOS DE HARDWARE

Procesador: Core

Memoria RAM: Mínimo: 1 Gigabytes (GB)

Disco Duro: 128Gb.

3.2. REQUERIMIENTOS MÍNIMOS DE SOFTWARE

Privilegios de administrador

Sistema Operativo: Windows XP en adelante, l'OS cualquier versión.



4. APLICATIVO CALISOFT

El objetivo del aplicativo es servir como herramienta para el proceso de evaluación de proyectos que se presenten en la Universidad de Cundinamarca, facilitando la labor del evaluador y ayudando a emplear una forma de evaluación más estándar y de mayor calidad, para sí, certificar la calidad y usabilidad misma del proyecto que se presente. Para acceder a la plataforma puede dar clic en el siguiente enlace:

<http://calisoft.pw>

El aplicativo cuenta con una extensión de Google Chrome que es necesaria en el módulo de Testing, esta extensión se puede descargar en el Dashboard del aplicativo.

4.1. ICONOS GENERALES

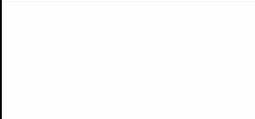
	Logo principal de la plataforma, representa el nombre de la mismo.
	Icono de la extensión, la extensión se puede encontrar en la página principal del aplicativo.

Tabla 49 Iconos generales

4.2. INGRESO AL APLICATIVO

Para el ingreso a la plataforma Calisoft es necesario que se dirija al siguiente link: <http://calisoft.com>. Una vez haya ingresado a la plataforma, el navegador debe mostrar la siguiente ventana;

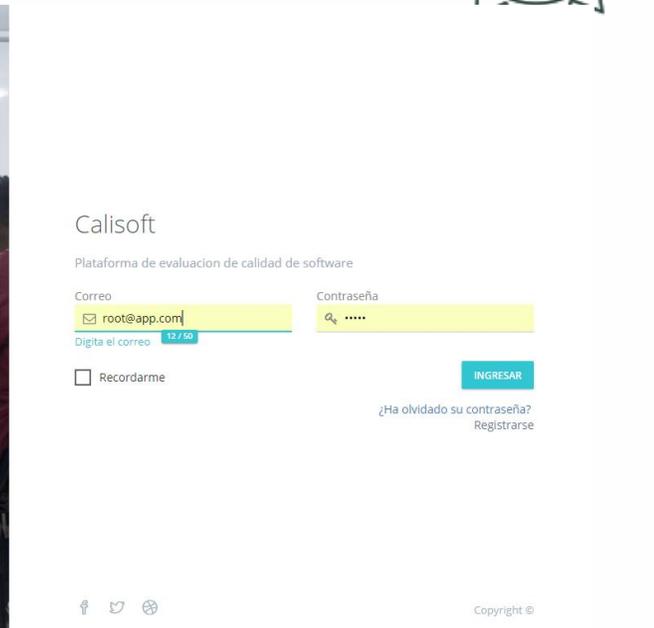


Imagen 1 Página principal

La imagen anterior muestra como es la página principal, la cual cuenta con una película de imágenes de las instalaciones de la universidad y con las opciones básicas de inicio de sesión.

4.2.1. Página principal

Los componentes que acompañan a la página principal se pueden apreciar en la siguiente imagen;



Imagen 2 Componentes página principal



En la parte izquierda superior se encuentra el logo del aplicativo, el cual corresponde al nombre de la plataforma, lleva consigo el color amarillo que representa uno de los colores de la Universidad de Cundinamarca.

En la parte derecha se encuentran los inputs o cuadros de texto, contraseña y correo, que corresponden al inicio de sesión, completar estos cuadros de textos es obligatorio si desea hacer uso de las funciones del aplicativo.

4.2.2. Inicio de sesión

Para realizar el inicio de sesión, la persona debe digitar su correo y contraseña en los campos que se muestran en la siguiente imagen;

Imagen 3 Inicio sesión

Si la persona cuenta ya con un perfil en la base de datos, deberá dar clic en el botón ingresar y el sistema lo redirigirá al módulo donde pertenezca su rol. En caso de que haya olvidado su contraseña, puede dar clic en el enlace “¿Ha olvidado su contraseña?”.



Imagen 4 Enlace recuperar contraseña

Una vez le haya dado clic, el sistema deberá redirigirlo a la siguiente vista;

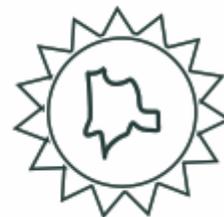
Imagen 5 Vista recuperar contraseña

En donde deberá escribir su correo y presionar en el botón enviar, el sistema enviará un mensaje al correo para restablecer su contraseña.

4.2.3. Registro

En caso de que sea una persona, en este caso nos referimos al desarrollador, no se encuentre registrado en la base de datos, debe dirigirse al enlace "Registrarse"

Imagen 6 Enlace de registro



Al dar clic en el enlace, el sistema lo redirigirá a la siguiente vista;

Calisoft

Plataforma de evaluación de calidad de software

Email

Nombre

Contraseña

Confirmar Contraseña

REGISTRARSE

Imagen 7 Vista de registro de usuario

El desarrollador debe diligenciar el formulario con su correo, nombre completo y una contraseña de más de 6 caracteres. Una vez diligenciado el formulario, debe dar clic en el botón “Registrarse” el sistema lo redirigirá al dashboard, donde procederá a registrar su proyecto.

4.3. USO DEL APLICATIVO

Por defecto, el sistema contará con tres usuarios creados por defecto pertenecientes a los tres roles que maneja el aplicativo. El administrador es el encargado de crear los usuarios evaluadores, mientras que los desarrolladores deberán registrarse mediante la página principal. A continuación, se narrará de una forma detallada todas las funciones que pueden hacer los diferentes usuarios dentro del aplicativo.

4.3.1. Administrador

El administrador es el usuario principal, encargado de realizar el cambio de los porcentajes de codificación.

Los registros de esta tabla son creados por defecto por el sistema, en sus filas se puede ver la descripción de cada uno de los componentes evaluativos y a su costado derecho cuenta con la opción de editar el registro correspondiente a la fila seleccionada.

Si se selecciona la opción de codificación, aparecerá la siguiente vista;



Items	Prioridad	Operación
Variables	5	
Clases	5	
Funciones	5	
Constantes	2	
Indentacion	3	
Comentarios	3	
Espacios De Nombre	2	

Descripción de los items

Editar regist

Imagen 8 Porcentajes de codificación

4.3.2. Evaluador

El evaluador es un usuario que crea el administrador, su propósito en la plataforma es el de calificar proyectos de software que se le asignen, cuando el administrador crea el usuario, un mensaje debe un mensaje debe llegar al correo del evaluador con la información general de su cuenta, una vez el evaluador conozca su contraseña, debe dirigirse a la página principal del sistema e ingresar su correo con su respectiva contraseña, el sistema validará la información y lo redirigirá a su dashboard;



Sección informativa sobre los porcentajes evaluativos en los

Sección informativa sobre los estándares de

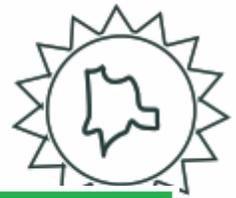
Proyectos

PROYECTOS ASIGNADOS	
prueba 1 <i>El proyecto todavía no se encuentra listo para evaluación</i>	
Nombre:	prueba 1
Estado:	ACTIVO
Categoría:	sistemas de información
Semillero	Linudec
Grupo de investigación:	GESAF
Creado el:	16/3/2018
Integranes:	FREDO
Evaluadores:	PAISA

Imagen 9 Dashboard evaluador

A la izquierda del dashboard del evaluador se encuentran dos secciones, la primera corresponde a toda la información correspondiente a los porcentajes evaluativos en los diferentes módulos que posee Calisoft, en la parte de abajo se encuentra toda la documentación sobre los estándares de codificación. Estas secciones de documentación ayudan al usuario a realizar una correcta evaluación sobre los proyectos que vaya a evaluar. En la parte derecha se visualizarán los proyectos que se le hayan asignado al evaluador.

Al dar clic en el enlace “Codificación” dentro de la sección de porcentajes, el sistema nos redirigirá a la siguiente vista;



Items	Prioridad
Variables	5
Clases	5
Funciones	5
Constantes	2
Indentacion	3
Comentarios	3
Espacios De Nombre	2

Imagen 10 Porcentajes de codificación evaluador

En donde el evaluador puede la prioridad de cada ítem que se maneja en el módulo de codificación, de esta forma sabrá que ítem tiene mayor prioridad que otros.

Al dar clic en el enlace “Codificación” dentro de la sección de Estándares, el sistema nos redirigirá a la siguiente vista;

Seccio →

<> ESTANDAR DE CODIFICACIÓN

La siguiente tabla que se presta a continuación es un manual de uso del estándar de codificación seleccionado por el semillero de “aplicaciones móviles” para el desarrollo de aplicativos web en la universidad de Cundinamarca, que utilicen el lenguaje de programación de PHP Se establecen 7 directrices de calificación que son los más relevantes en el momento de escribir código entre las cuáles se presentan a continuación

Variables	Clases	Funciones	Constantes	Indentación	Comentarios
-----------	--------	-----------	------------	-------------	-------------

Espacios De Nombre

La notación recomendada para la declaración de variables es la notación lowerCamelCase está consiste en que cuando las variables formadas por más de una palabra cada una de estas deberá iniciar en mayúscula exceptuando la primera palabra, si la palabra conlleva guiones bajos en una posición diferente a la inicial la declaración será errónea, también la longitud mínima de una variable es de 3 caracteres exceptuando las variables \$i, \$j, \$k que se utilizan en los ciclos “for”.

Imagen 11 Documentación de los estándares de codificación

Volviendo a la página principal del evaluador y en el apartado de proyectos se escoge la opción de “CODIFICACIÓN” como se ve en la imagen 40, el sistema redirigirá al evaluador a la siguiente vista;



Imagen 12 Tabla de Archivos de codificación del evaluador

Dentro del módulo para la evaluación de codificación, luego que el estudiante haya subido la documentación requerida para la posterior calificación los archivos se desplegarán de la siguiente manera

El primer paso es la información que trae la vista para realizar la correcta calificación de los archivos de codificación o Scripts



Imagen 13 Modal ayuda de lista de Scripts

En el modal se encuentran las funciones de la evaluación de codificación, también un botón para cerrar el modal



Volviendo al listado de archivos de codificación, se encuentra una barra de búsqueda para filtrar por el nombre de los archivos, el segundo paso es realizar la calificación de los Scripts, para esto se tiene que oprimir el botón amarillo el cual se redirigirá a la siguiente ventana

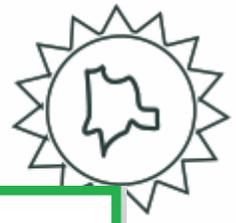
The screenshot shows a web application interface for evaluating a PHP script. On the left, a code editor displays PHP code for a `UserController` class. On the right, a table shows evaluation metrics for various items. A 'Modal De ayuda' (Help Modal) is open, and a 'Calificación' (Evaluation) button is highlighted. A 'Crear' (Create) button is also visible. A 'Calificación' button is located below the table. A 'Aceptar Calificación' (Accept Evaluation) button is at the bottom right. A 'Calificación Automática' (Automatic Evaluation) button is at the top right. A 'Visualización del código fuente' (Source Code Visualization) button is on the left. A 'Calificación' button is also present in the top right corner of the interface.

Items	Total Del Item	Aprobados Del Item	Calificación
Variables	0	0	0
Clases	0	0	0
Funciones	0	0	0
Constantes	0	0	0
Identación	0	0	0
Comentarios	0	0	0
Espacios De Nombre	0	0	0

Comentarios

Imagen 14 Panel de calificación de Scripts

El primer paso es abrir el modal de ayuda para la guía correcta de la evaluación del script seleccionado



Ayuda Del Módulo Para La Evaluación De Archivos De Codificación

Funciones Del Módulo Para La Evaluación De Archivos De Codificación

Pasos para realizar la correcta evaluación del archivo de codificación correspondiente al proyecto

- En caso de no haber leído el estándar de codificación, se recomienda dar click al botón en la parte de debajo de la visualización del código, este desplegara otra ventana mostrando un archivo PDF con el estándar de programación
- Existen 2 métodos de calificación, si escoge **calificación automática** saldrá un mensaje de que el Script ya ha sido evaluado por lo consiguiente está a su criterio si realizar un comentario o no, si desea revisar los puntajes obtenidos por el script refrescar la página o apretar "F5", luego apretar el botón de "realizar calificación" para dirigirse a la tabla de scripts.
- Si escoge **la calificación manual** es importante determinar los ítems que hay en la visualización del código, por lo tanto puede llenar la tabla de manera manual para relizar la correspondiente calificación del código fuente, está en su criterio realizar comentario del archivo o no, por ultimo apretar el botón de "realizar calificación" para dirigirse a la tabla de scripts.

Importante

CERRAR

Imagen 15 Modal de ayuda para Calificar un Script

Volviendo al panel de evaluación del script, se puede visualizar el código a la parte izquierda de la ventana para una completa información sobre la que se evalúa, luego es importante escoger si una evaluación automática o una evaluación manual, para la evaluación automática, presionar el botón que dice calificación automática

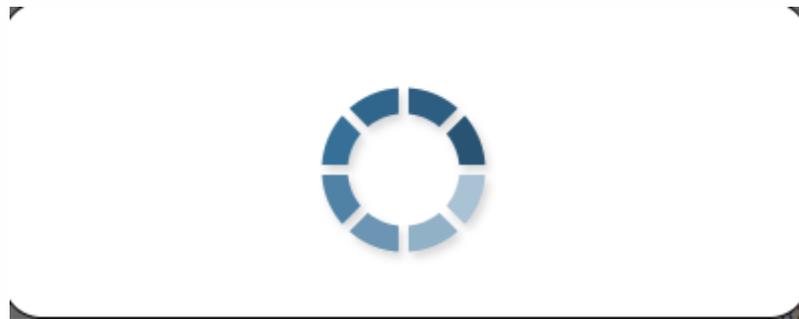


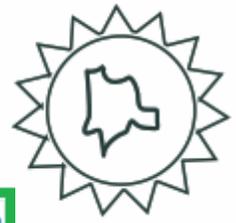
Imagen 16 Ejemplo Block UI

Se mostrará una herramienta de bloqueo de interfaz mientras que envía la petición para realizar la calificación automática, luego de la espera saldrá un mensaje diciendo que el Script ha sido calificado exitosamente, no obstante, la tabla de calificación manual no se modificara automáticamente

Items	Total Del Item	Aprobados Del Item	Calificación
Variables	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Clases	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Funciones	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Constantes	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Identacion	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Comentarios	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Espacios De Nombre	<input type="text" value="0"/>	<input type="text" value="0"/>	0

Imagen 17 Panel de Calificación Manual

Si desea observar los resultados solo es recargar la página, no obstante, ya se ha calificado el script, queda bajo criterio si desea editar la calificación de los estándares de codificación manualmente



Items	Total Del Item	Aprobados Del Item	Calificación
Variables	4	4	1
Clases	Total de items	1	1
Funciones	12	Total de items aprobados	Promedio para la calificacion
Constantes	0		0
Identacion	13	11	0.85
Comentarios	14	5	0.36
Espacios De Nombre	1	0	0

Imagen 18 Panel de Calificación Manual con Resultados

Luego de recargar la página se mostrarán automáticamente los resultados arrojados por la calificación automática, para realizar cambios o realizar la calificación manual solo es cambiar las cajas de texto del total de ítems y cuantos aprobó en el script, automáticamente general el promedio de la división y guarda en base de datos

Comentario

Campo para la realizacion de un comentario

ACEPTAR CALIFICACIÓN

Boton de aceptar calificacion y guardar

Imagen 19 Panel de comentarios

Para terminar la evaluación del Script es importante determinar si hay falencias en el archivo, por ende, existe la alternativa de creación de un comentario para hacer saber al desarrollador al que se evalúa cuáles son las correcciones, por ultimo solo es oprimir el botón de aceptar calificación, el cual mostrara de nuevo la tabla con



los archivos de codificación que hacen falta para la evaluación y cambiando el botón de calificación a la muestra de ítems calificados por orden

Nombre Del Archivo	Operaciones
6290_UserController.php	
5568_TiposDocumentoController.php	
3567_TestInputsController.php	
3671_TestingController.php	

Script calificado

Script sin

Imagen 20 Diferencia entre Script calificado y sin calificar

Ahí se determina cual script ya está calificado y ya cual falta por calificar

Al presionar el botón azul de ver la calificación de cada ítem se deben tener en cuenta 2 cosas

Cuando no se realiza una calificación automática el panel aparecerá vacío como se muestra en la siguiente imagen

Item	Atributo	Fila	Aprobado
------	----------	------	----------

Imagen 21 Tabla de Ítems sin Evaluar Evaluador

Aunque solo es la muestra de lo que se calificó no afecta en la nota del estudiante, cuando se califica de manera autónoma el analizador léxico guarda en base de datos los atributos y los califica por eso se muestran en la siguiente imagen



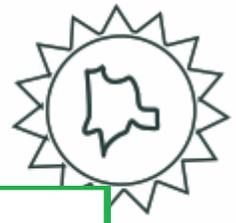
UserController.php

Item	Atributo	Fila	Aprobado
Variables	users	16	✓
Variables	request	30	✓
Variables	user	44	✓
Variables	usuario	60	✓
Clases	UserController	14	✓
Funciones	index	30	✓
Funciones	store	41	✓
Funciones	destroy	60	✓
Funciones	proyectos	68	✓
Funciones	searchFreeStudents	82	✓
Funciones	searchEvaluators	87	✓
Funciones	invitaciones	92	✗

Imagen 22 Tabla de ítems Evaluados

En el modal se muestran los ítems que se evaluó, el atributo evaluado, en que fila se encuentra y si está aprobado o rechazado

Volviendo otra vez a la tabla de los archivos de codificación, se deben calificar todos los archivos ya sean manual o automático, como adición a la vista del evaluador se muestra también el reporte generado por todos los scripts al oprimir el botón de los reportes en la parte superior



UDEC
UNIVERSIDAD DE
CUNDINAMARCA

CALISOFT

Calle 14 con Avenida 15
Universidad de Cundinamarca - Ext. Facatativá
(+57 1) 892 0706 | 892 0707
unicundi@ucundinamarca.edu.co

modulo para la presentacion de manuales - Evaluación de codificación

6290_UserController.php			
Nombre	Total	Acertadas	Nota
Variables	4	4	1
Clases	1	1	1
Funciones	12	6	0.5
Constantes	0	0	0
Identación	13	11	0.85
Comentarios	14	5	0.36
Espacios De Nombre	1	0	0
Observación	script calificado correctamente		

Imagen 23 Reporte del módulo para la evaluación de codificación Evaluador

el reporte generado por todos los scripts, con las observaciones del evaluador, al final del reporte se muestra el porcentaje del índice de cohesión o aceptación que tiene el proyecto frente al módulo para la evaluación de estándares de codificación

Promedio General (Codificación)

71.5%

Imagen 24 Nota final calculando el índice de cohesión o aceptación evaluador

Una vez se hayan realizado todas las evaluaciones en los diferentes módulos, el evaluador cuenta con dos opciones sobre el proyecto;



Prueba 1		EVALUACIÓN ▾
Nombre:	Prueba 1	
Estado:	EVALUACION	
Categoría:	sistemas de información	
Semillero	Linudec	
Grupo de investigación:	GESAF	
Creado el:	7/3/2018	
Integrantes:	FREDO	
Evaluadores:	PAISA	
VOLVER A ACTIVAR PROYECTO		TERMINAR EVALUACIÓN

Imagen 25 Proceso final de calificación de proyectos

Una de ellas es habilitar el proyecto para que el estudiante pueda hacer las correcciones que el evaluador le haya puesto, para ello debe dar clic en el botón “VOLVER A ACTIVAR EL PROYECTO”, o finalizar el proceso de evaluación dejando la calificación que haya obtenido, para ello debe dar clic en el botón “TERMINAR EVALUACIÓN”, al oprimir este botón se finaliza el proceso de evaluación y el estudiante será notificado y podrá ver sus resultados en su dashboard.

4.3.3. Desarrollador

El desarrollador es el usuario encargado de subir toda la documentación referente a su proyecto a la plataforma, la documentación engloba diagramas, bases de datos, código fuente y estructura de los formularios que emplee su proyecto, el sistema guiará al desarrollador para que realice un correcto uso de la plataforma. Al momento en que el desarrollador se registre en la página principal, el sistema lo redirigirá a la siguiente vista;

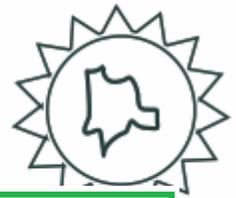


Imagen 26 Dashboard del desarrollador

Las secciones informativas de estándares y porcentajes corresponden a las mismas vistas explicadas anteriormente en el apartado del evaluador. El desarrollador como paso inicial, deberá crear su proyecto, para ello puede dar clic en el apartado izquierdo en el icono de un lápiz, o en el enlace que se encuentra en zona central de la página principal, ambos hipervínculos redirigirán al desarrollador a la siguiente vista;



The image shows a web form titled "REGISTRO PROYECTO" with a green border. The form contains the following fields and a button:

- Nombre del proyecto:** A text input field with a red callout box pointing to it from the left.
- Categoría:** A dropdown menu with "sistemas de información" selected. A red callout box points to it from the right.
- Semillero:** A dropdown menu with "Linudec" selected. A red callout box points to it from the left.
- Grupo de investigación:** A dropdown menu with "GESAF" selected. A red callout box points to it from the right.
- ENVIAR:** A blue button with white text. A red callout box points to it from the left.

Imagen 27 Registro del proyecto

El nombre del proyecto debe ser único y debe corresponder al nombre del proyecto que vaya a registrar, en la sección de categoría debe seleccionar a qué categoría pertenece su proyecto, al igual que en la sección de semillero y grupo de investigación, al haber digitado el nombre del proyecto y asignado los grupos a los que pertenezca, debe dar clic en el botón enviar para crear su proyecto. Una vez creado la página principal de su dashboard se vera de la siguiente manera;

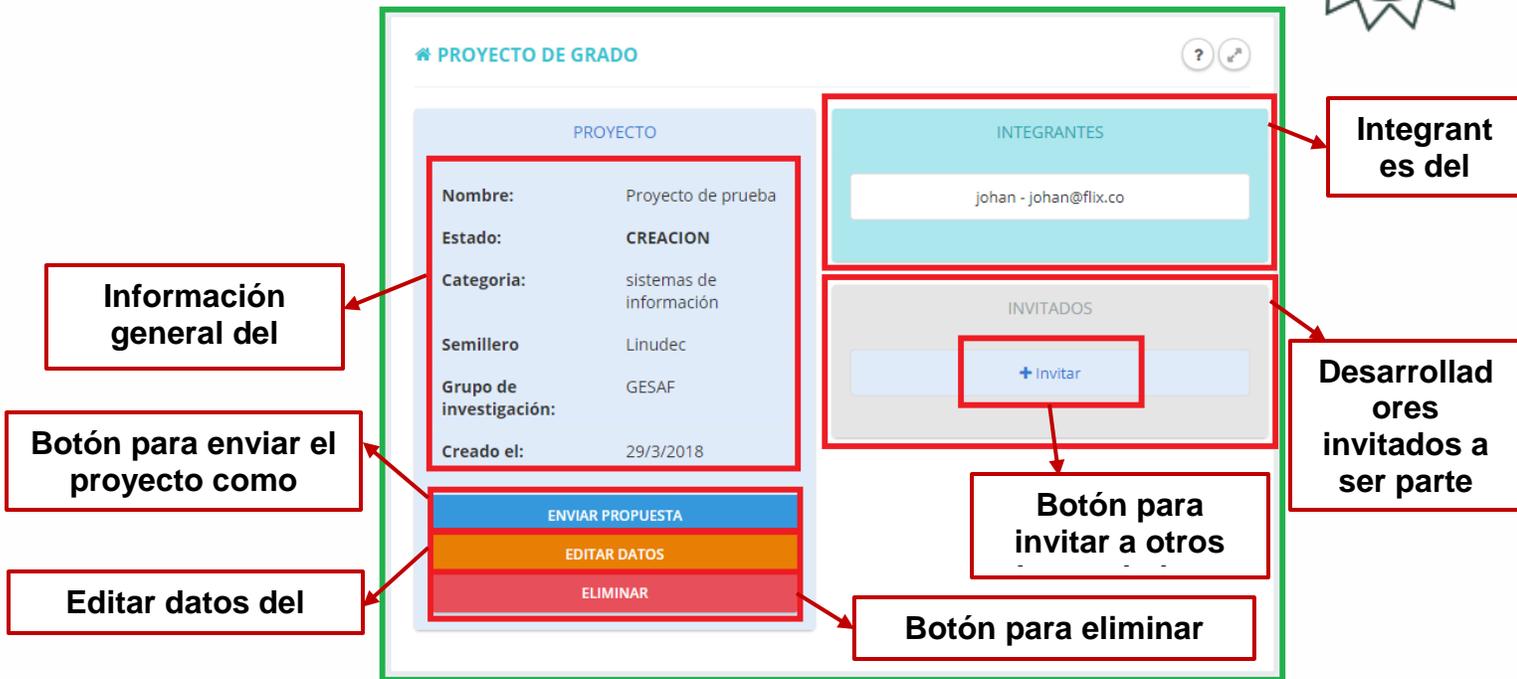
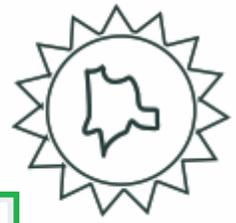


Imagen 28 Proyecto listo para enviarlo como propuesta

En la sección superior izquierda se encuentran los datos generales del proyecto, y abajo se encuentran una serie de botones que corresponden a; enviar el proyecto como propuesta con los datos que se aprecian en la parte de arriba, editar los datos del proyecto y eliminar proyecto. En la parte superior derecha se encuentra una sección que corresponde a los desarrolladores que forman parte del proyecto, si usted tiene un compañero desarrollador, deberá dar clic en el botón invitar que se encuentra en la sección de abajo. Una vez dado clic se desplegará el siguiente Modal;

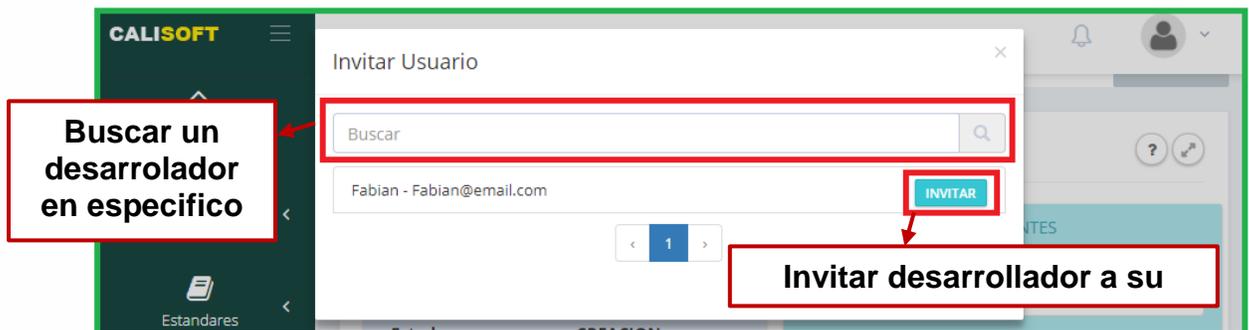


Imagen 29 Modal invitar usuario

En donde podrá buscar en particular a un usuario desarrollador que se encuentre registrado en la plataforma y dando clic en el botón "INVITAR" sobre la fila del



usuario que desee invitar, el sistema enviará una notificación al usuario invitado y su página principal se actualizará de la siguiente forma;

The screenshot shows a web interface for a 'PROYECTO DE GRADO'. On the left, under the 'PROYECTO' section, the following details are listed: Nombre: Proyecto de prueba; Estado: CREACION; Categoría: sistemas de información; Semillero: Linudec; Grupo de investigación: GESAF; Creado el: 29/3/2018. Below this are three buttons: ENVIAR PROPUESTA (blue), EDITAR DATOS (orange), and ELIMINAR (red). On the right, under the 'INTEGRANTES' section, the user 'johan - johan@flix.co' is listed. Below that, under the 'INVITADOS' section, the user 'Fabian - Fabian@email.com' is listed with a '+ Invitar' button. A red box highlights the 'INVITADOS' section, and an arrow points from the '+ Invitar' button to a separate box labeled 'Usuarios'.

Imagen 30 Usuarios invitados

De esa forma se puede apreciar los usuarios desarrolladores que se encuentran invitados al proyecto, a los usuarios a quienes se les envíen notificaciones, se les mostrará la siguiente vista en la sección de notificaciones;

The screenshot shows a notification interface titled 'INVITACIONES'. The notification text reads: 'El proyecto Proyecto de prueba, a enviado una solicitud, para que haga parte del equipo de trabajo.' To the right of the notification are two buttons: 'ACEPTAR' (blue) and 'RECHAZAR' (grey). A red box labeled 'Aceptar invitación al' has an arrow pointing to the 'ACEPTAR' button. Another red box labeled 'Rechazar' has an arrow pointing to the 'RECHAZAR' button.

Imagen 31 Aceptar invitación



En donde el usuario invitado tiene la decisión de incorporarse al proyecto al que se le invito, o no, si escoge la opción de aceptar la invitación, automáticamente aparecerá el proyecto en su página principal y la vista al usuario que envió la invitación se actualizará de la siguiente manera;

The screenshot displays a web interface for a project titled "PROYECTO DE GRADO". The interface is divided into two main sections: "PROYECTO" and "INTEGRANTES".

PROYECTO Section:

- Nombre:** Proyecto de prueba
- Estado:** CREACION
- Categoría:** sistemas de información
- Semillero:** Linudec
- Grupo de investigación:** GESAF
- Creado el:** 29/3/2018

Below the project details are three buttons: "ENVIAR PROPUESTA" (blue), "EDITAR DATOS" (orange), and "ELIMINAR" (red).

INTEGRANTES Section:

- Members listed: Fabian - Fabian@email.com and johan - johan@flix.co

INVITADOS Section:

- Contains a "+ Invitar" button.

A red box highlights the "INTEGRANTES" section, and a red arrow points from this box to a separate box labeled "Sección de integrantes".

Imagen 32 Sección de integrantes actualizada

Informando qué usuarios hacen parte del proyecto. Al verificar los datos del proyecto y sus integrantes, el desarrollador debe dar clic en el botón "ENVIAR PROPUESTA", el sistema notificará al administrador y este procederá a aceptar o rechazar el proyecto, si el proyecto es aceptado, el administrador procederá a asignar evaluadores al proyecto y la vista principal del desarrollador se actualizará de la siguiente forma;

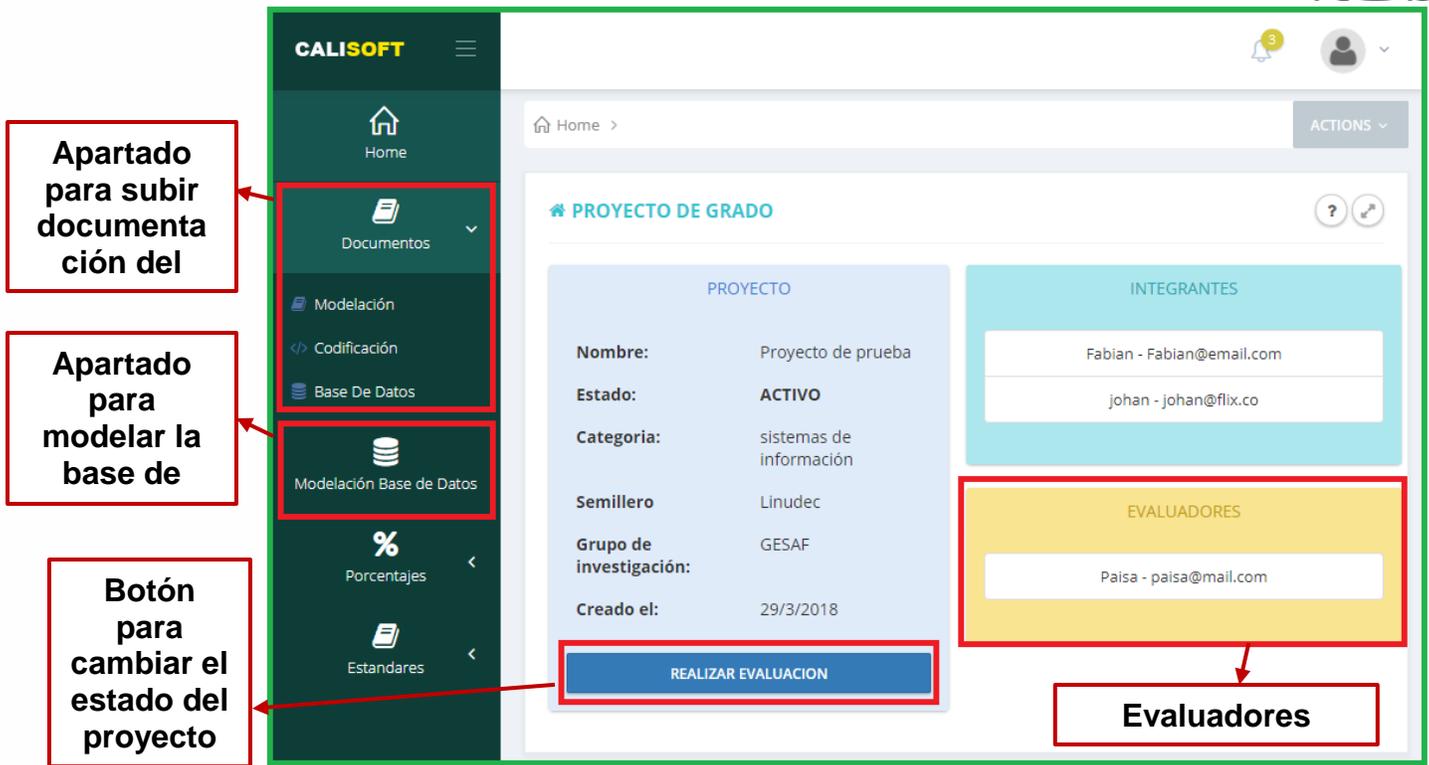
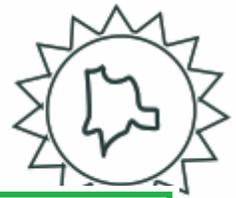


Imagen 33 Proyecto listo para evaluación

En donde se podrá ver los evaluadores asignados, al lado izquierdo se habilitarán las secciones de documentación y modelación de base de datos, y en la parte central se activará un botón para modificar el estado del proyecto a “Evaluación” una vez modificado el estado, ya no se podrá cambiar el estado ni modificar la documentación. Como primer paso, el desarrollador debe subir los archivos correspondientes al módulo para la evaluación de estándares de codificación, para ello debe ir a la sección de documentos y dar clic en “Codificación” y el sistema lo redirigirá a la siguiente vista



Imagen 34 Primera Vista del módulo de codificación



En la vista se encuentra una barra de búsqueda de archivos que ayuda al desarrollador a filtrar la búsqueda por una palabra, también encontrara un modal de ayuda que al presionar se desplegara un menú de ayuda y un botón de cerrar para cerrar el modal

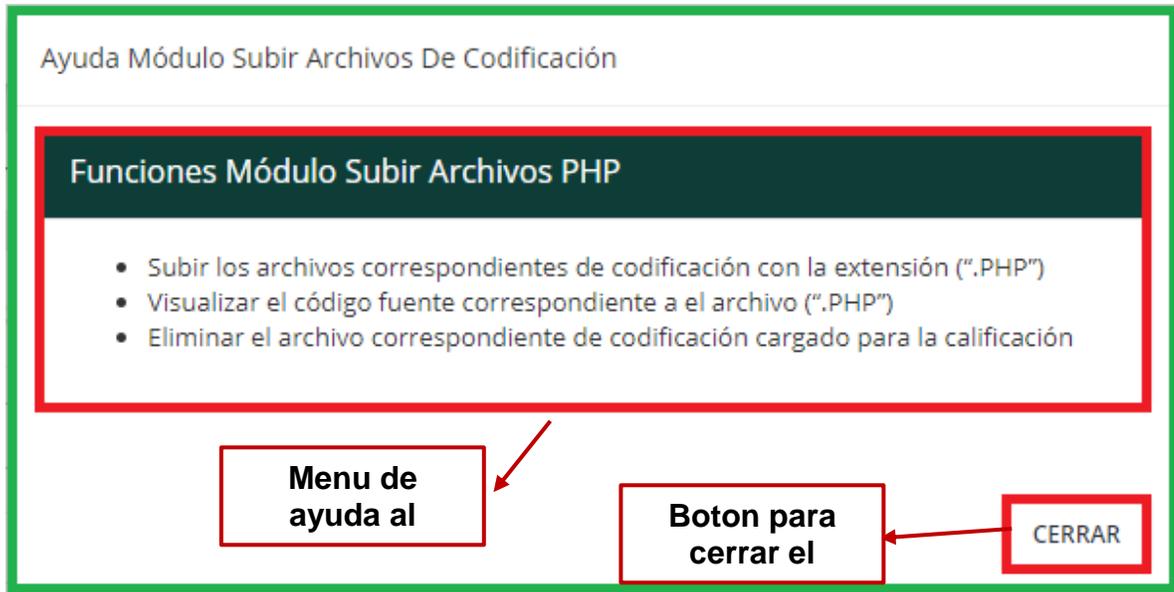


Imagen 35 Modal de ayuda para subir archivos de codificación

Volviendo a la vista actual para subir los archivos correspondientes a él módulo de codificación es oprimir el botón de subir documentos, luego se desplegará un dropzone con la siguiente ventana



Imagen 36 Panel de carga para subir Archivos de codificación

En este caso se puede arrastras los archivos con la extensión (.PHP) o dando click en el dropzone se despegará una ventana de búsqueda para los archivos, luego

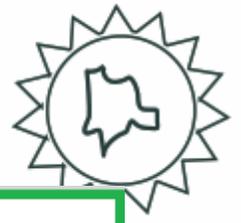


Imagen 37 Panel de Carga con Archivos de codificación

Luego de haber cargado los archivos de codificación correctamente se desplegará una tabla con el listado de archivos

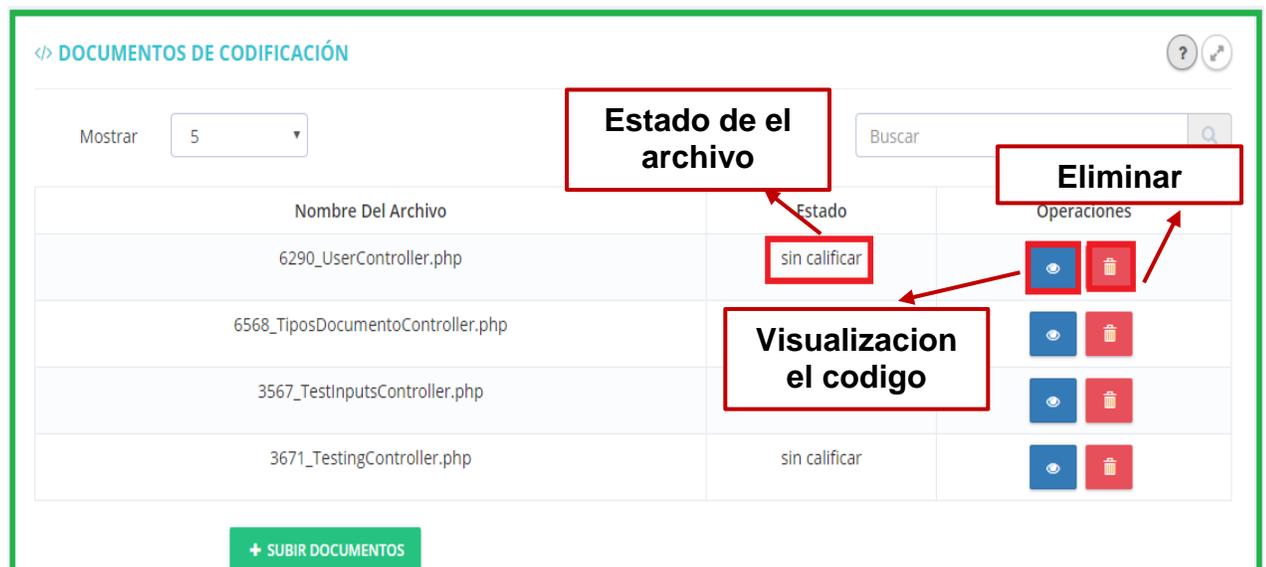


Imagen 38 tabla de listado de archivos de codificación estudiante



El estado del archivo cambia respectivamente luego que el profesor asignado se encargue de realizar la calificación del script, existen 2 operaciones para realizar, con el documento, la primera es eliminar el script para no ser calificado, en caso de subirlo erróneamente, se desplegara la siguiente ventana

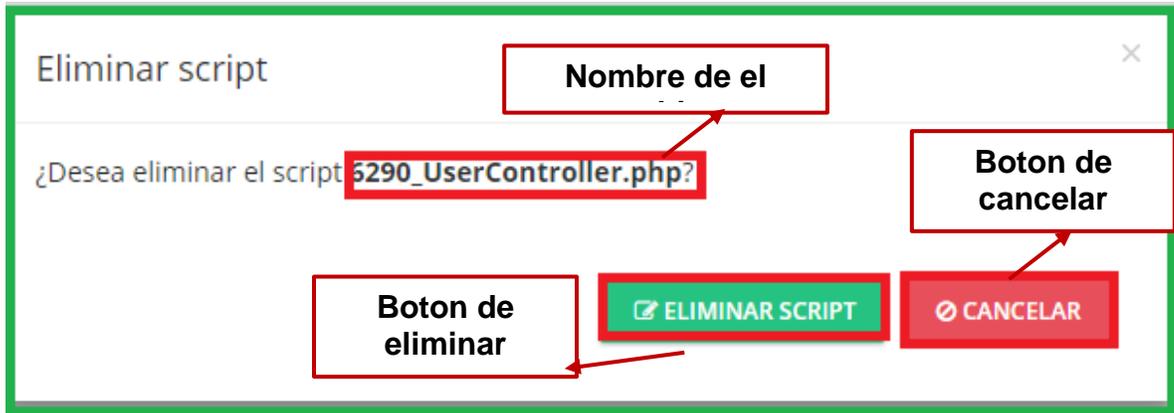


Imagen 39 Modal para eliminar un Script

Trae el nombre del archivo para verificar el archivo seleccionado sea correcto, en el modal hay 2 botones eliminar script correctamente o cancelar acción y volver a la tabla de scripts

Volviendo a la tabla de scripts, la segunda operación es el modal de visualización del código fuente, presionando el botón azul con el ojo se desplegará la siguiente vista

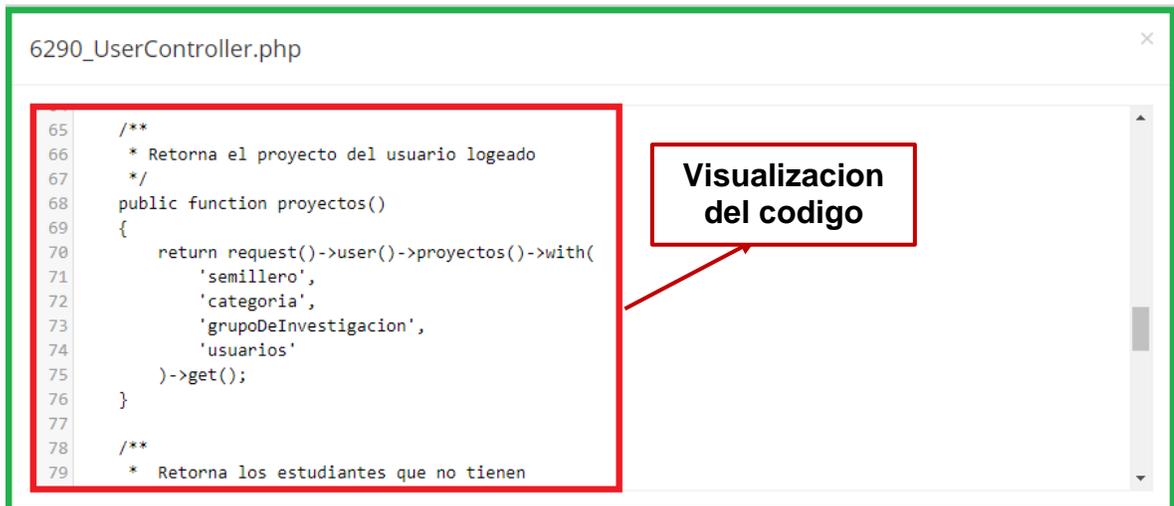


Imagen 40 Panel de Visualización del código del desarrollador

Una vez subida toda la documentación del proyecto, el desarrollador debe proceder a la página principal de su dashboard y dar clic en el Botón “REALIZAR



EVALUACIÓN”, de esta forma se cambia el estado del proyecto, el evaluador es notificado y procederá a revisar toda la documentación que el desarrollador subió al sistema. Tenga en cuenta que una vez modificado el estado del proyecto no podrá realizar ningún cambio a la documentación. La vista principal del desarrollador se actualizará de la siguiente forma;

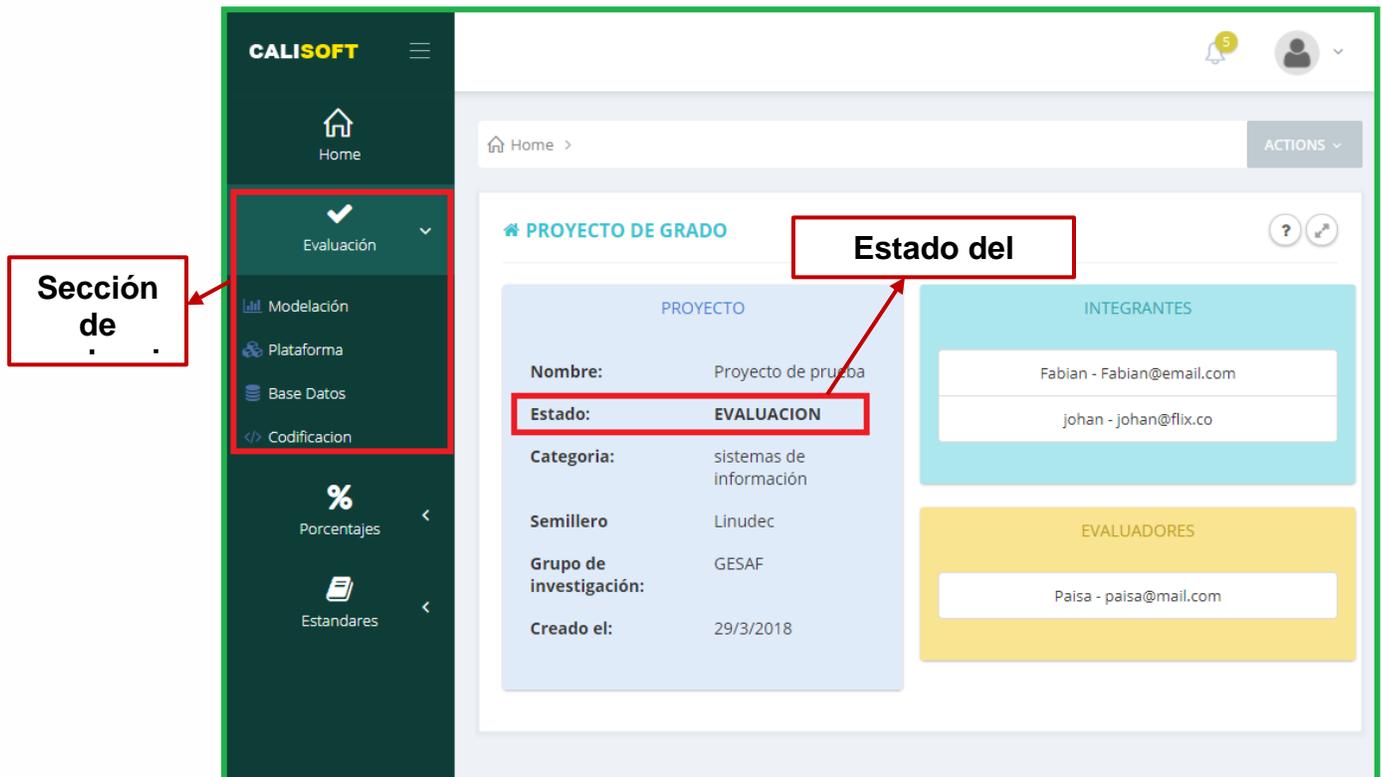
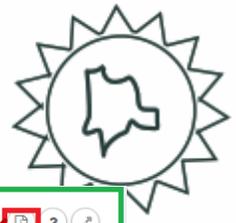


Imagen 41 Proyecto en estado de evaluación

Modificando el estado a “EVALUACIÓN” y agregando una sección de evaluación, donde el desarrollador podrá ver los resultados de la evaluación del evaluador sobre su documentación. Para ver los resultados de los resultados de la codificación, el evaluador debe dirigirse a la sección de evaluación, dar clic en el apartado de “Codificación” y el sistema lo redirigirá a la siguiente vista



Nombre Del Archivo	Operaciones
6290_UserController.php	
6568_TiposDocumentoController.php	
3567_TestInputsController.php	
3671_TestingController.php	

Imagen 42 Panel de Archivos de codificación calificados en la vista del desarrollador

En el panel de scripts se observan los archivos que se calificaron, se debe tener en cuenta 2 casos, el primero que el evaluador haya calificado de manera manual el script por lo que el modal aparecería vacío, no es de preocuparse

Item	Atributo	Fila	Aprobado
------	----------	------	----------

Imagen 43 Tabla de Ítems sin Evaluar en la vista del desarrollador

No aparecerá nada ya que el analizador léxico no se ejecutó y por ende no se llenó la base de datos con los ítems del código el segundo caso, que el evaluador haya realizado una calificación automática por lo que al presionar el botón azul aparecerá este modal

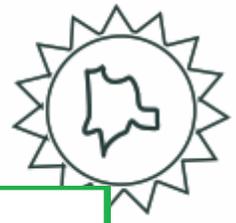


UserController.php			
Item	Atributo	Fila	Aprobado
Variables	users	16	✓
Variables	request	30	✓
Variables	user	44	✓
Variables	usuario	60	✓
Clases	UserController	14	✓
Funciones	index	30	✓
Funciones	store	41	✓
Funciones	destroy	60	✓
Funciones	proyectos	68	✓
Funciones	searchFreeStudents	82	✓
Funciones	searchEvaluators	87	✓
Funciones	invitaciones	92	✗

Imagen 44 Tabla de Ítems evaluados en la vista del desarrollador

En el modal se muestran los ítems que se evaluó, el atributo evaluado, en que fila se encuentra y si está aprobado o rechazado

se muestra también el reporte generado por todos los scripts al oprimir el botón de los reportes en la parte superior de la tabla donde se muestra el listado de los archivos de codificación



UDEC
UNIVERSIDAD DE
CUNDINAMARCA

CALISOFT

Calle 14 con Avenida 15
Universidad de Cundinamarca - Ext. Facatativá
(+57 1) 892 0706 | 892 0707
unicundi@ucundinamarca.edu.co

modulo para la presentacion de manuales - Evaluación de codificación

Nombre	6290_UserController.php		
Items	Total	Acertadas	Nota
Variables	4	4	1
Clases	1	1	1
Funciones	12	6	0.5
Constantes	0	0	0
Identación	13	11	0.85
Comentarios	14	5	0.36
Espacios De Nombre	1	0	0
Observación	script calificado correctamente		

Imagen 45 Reporte del módulo para la evaluación de codificación Desarrollador

el reporte generado por todos los scripts, con las observaciones del evaluador, al final del reporte se muestra el porcentaje del índice de cohesión o aceptación que tiene el proyecto frente al módulo para la evaluación de estándares de codificación

Promedio General (Codificación)

71.5%

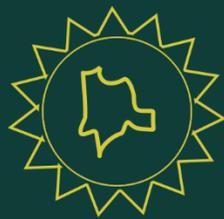
Imagen 46 Nota final calculando el índice de cohesión o aceptación Desarrollador

Para ver los resultados del modelamiento de la base de datos, el desarrollador debe dirigirse a la sección de evaluación, dar clic en el apartado de “Base de Datos” y el sistema lo redirigirá a la siguiente vista;

5. CONTROL DE CAMBIOS DEL MANUAL



Actualización Nro.	Descripción del cambio	Versión del aplicativo	Fecha de cambio



www.unicundi.edu.co
unicundi@mail.unicundi.edu.co
Línea gratuita 018000 976000



GP-CER365641



CO-SC-CER395037



SO-CER365037

Dirección de Sistemas y Tecnología
sistemasytecnologia@mail.unicundi.edu.co
PBX: 828 14 83 Ext. 110
Sede Fusagasugá

Anexo B. Manual de instalación.

MANUAL DE INSTALACIÓN

CONTENIDO

INTRODUCCIÓN	168
REQUERIMIENTOS MÍNIMOS DEL SISTEMA	169
PROGRAMAS NECESARIOS	169
COMPOSER	169
XAMMP	171
NODE.JS.....	174
GOOGLE CHROME	175
INSTALACIÓN	176
Uso de XAMPP para ubicar la carpeta del proyecto	176
Instalación de Composer en la carpeta del proyecto.....	178
Creación de la base de datos.....	182
Modificación del archivo example.env	183
Configuración del repositorio de MAIL	186
Configuración de Pusher para notificaciones en tiempo real	190
Cambiar nombre archivo exmaple.env	195
Llave de acceso	197
Creación ruta a carpeta Storage	197
Instalación de Vue de Node.js en la carpeta del proyecto.....	197
Migraciones	199
Seeders	201
Correr el aplicativo	202
Acceso al login	202
REFERENCIAS	203

ILUSTRACIONES

Ilustración 1	Página principal de Composer.....	169
Ilustración 2	Documentación de instalación de Composer	170
Ilustración 3	Instalación de Composer	171
Ilustración 4	Página principal de XAMPP.....	172
Ilustración 5	Recomendaciones antes de la instalación de XAMPP	173
Ilustración 6	Proceso de la instalación de XAMPP	173
Ilustración 7	Página principal de node.js.....	174
Ilustración 8	Proceso de instalación de Node.js.....	175
Ilustración 9	Página de descarga de Google Chrome.....	176
Ilustración 10	Ruta de la carpeta principal de Calisoft	176
Ilustración 11	Interfaz XAMPP	177
Ilustración 12	Interfaz XAMPP con servicios activados	178
Ilustración 13	Ventana ejecutar.....	178
Ilustración 14	Símbolo del sistema.....	179
Ilustración 15	Ejecución del comando "cd.."	179
Ilustración 16	Ejecución del comando "cd.."	179
Ilustración 17	Cmd ubicación de la carpeta principal	180
Ilustración 18	Composer install en CMD	180
Ilustración 19	Procedimiento de la instalación de Composer en carpeta principal	180
Ilustración 20	Finalización de Composer en carpeta principal	181
Ilustración 21	Comando Composer update en CMD.....	181
Ilustración 22	Actualización Composer en CMD	182
Ilustración 23	Creación base de datos.....	183
Ilustración 24	Ubicación archivo "example.env"	184
Ilustración 25	Contenido archivo "example.env"	185
Ilustración 26	Configuración de MAIL	186
Ilustración 27	Página principal de Mailtrap	186
Ilustración 28	Inicio de sesión Mailtrap	187
Ilustración 29	Backend Mailtrap	187
Ilustración 30	Administración de mensajes de Mailtrap	188
Ilustración 31	Apartado Integrations Mailtrap.....	189
Ilustración 32	Atributos de Mailtrap en Laravel	190
Ilustración 33	Apartado de Mail en archivo example.env	190
Ilustración 34	Página principal de Pusher	191
Ilustración 35	Welcome to Pusher	192
Ilustración 36	Modificación de Pusher	193
Ilustración 37	Backend Pusher	194

Ilustración 38 Pestañas Pusher	194
Ilustración 39 App Keys Pusher	195
Ilustración 40 Configuración Pusher en archivo example.env	195
Ilustración 41 Cambiar nombre del archivo example.env	196
Ilustración 42 Archivo. env	196
Ilustración 43 Php artisan key:generate	197
Ilustración 44 Proceso exitoso del comando Php artisan storage:link	197
Ilustración 45 Proceso de instalación de Vue	198
Ilustración 46 Finalización exitosa de Vue en carpeta del proyecto	199
Ilustración 47 Comando php artisan migrate	200
Ilustración 48 Migraciones exitosas	200
Ilustración 49 Comando php artisan db:seed exitoso	201
Ilustración 50 Registros de la tabla TBL_Usuarios	201
Ilustración 51 Datos de los seeders	202
Ilustración 52 Ip del servidor local	203
Ilustración 53 Página principal de Calisoft	203

INTRODUCCIÓN

Este documento es una guía por la cual el técnico que dará soporte a la aplicación Calisoft, debe tener en cuenta a la hora de realizar la precisa instalación para que todos los componentes y extensiones del aplicativo

funcionen correctamente. A continuación, se darán a conocer los programas necesarios para este proceso.

REQUERIMIENTOS MÍNIMOS DEL SISTEMA

PROGRAMAS NECESARIOS

COMPOSER

“Composer es una herramienta para la administración de dependencias en PHP. Le permite declarar las bibliotecas de las que depende su proyecto y las administrará (las instalará / actualizará).”(Composer, n.d.) Para realizar la instalación es necesario dirigirse a la página; <https://getcomposer.org/> , en la ilustración 1 podrá observar como es la interfaz.

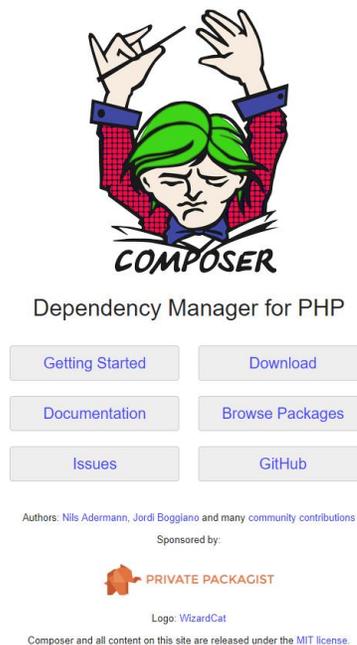


Ilustración 1 Página principal de Composer

Dentro de la interfaz, usted deberá seleccionar la opción de “Download” y se redirigirá a la siguiente página;

Download Composer

Windows Installer

The installer will download composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('SHA384', 'composer-setup.php') === '544e09ee996cdf60ece3804abc52599c22b1f40f4323403c44d4
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

This installer script will simply check some php.ini settings, warn you if they are set incorrectly, and then download the latest composer.phar in the current directory. The 4 lines above will, in order:

- Download the installer to the current directory
- Verify the installer SHA-384 which you can also [cross-check here](#)
- Run the installer
- Remove the installer

WARNING: Please do not redistribute the install code. It will change with every version of the installer. Instead, please link to this page or check [how to install Composer programmatically](#).

Installer Options

`--install-dir`

You can install composer to a specific directory by using the `--install-dir` option and providing a target directory. Example:

Ilustración 2 Documentación de instalación de Composer

Se procede a dar clic en el hipervínculo “Composer.Setup.exe” y se iniciará la descarga automáticamente. Una vez descargado el programa, se debe ejecutar en modo administrador para evitar errores de instalación y aparecerá la siguiente ventana;

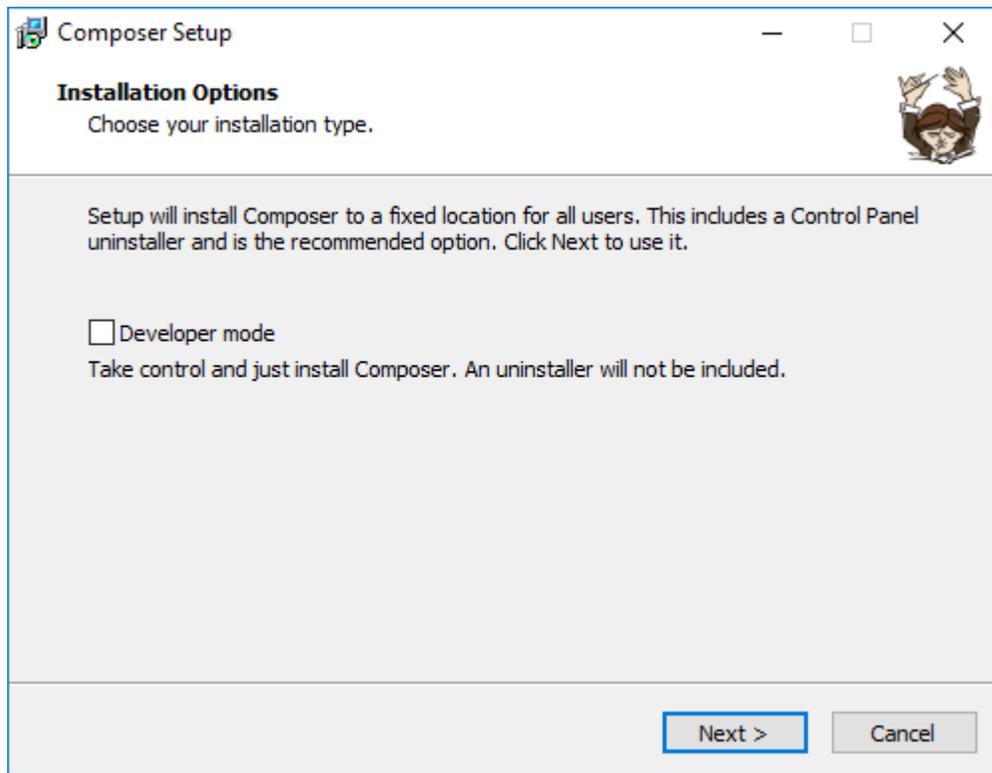


Ilustración 3 Instalación de Composer

Es necesario que omita la opción de “Developer mode” y proceda a pulsar el botón “Next”. Debe realizar el mismo procedimiento en las siguientes ventanas hasta que finalice la instalación.

Una vez finalizada la instalación se cierra la ventana y de esta forma Composer queda instalado en el ordenador.

XAMPP

“El objetivo de XAMPP es crear una distribución fácil de instalar para desarrolladores que se están iniciando en el mundo de Apache. XAMPP viene configurado por defecto con todas las opciones activadas”. (Apache Friends, 2015) Para proceder a su instalación debe ingresar a la siguiente página; <https://www.apachefriends.org/es/index.html> que corresponde a la página principal de XAMPP

Apache Friends

Descargar Complementos Alojamiento Comunidad Acerca de  ES

XAMPP Apache + MariaDB + PHP + Perl

¿Qué es XAMPP?

XAMPP es el entorno más popular de desarrollo con PHP

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.



Descargar
Pulsa aquí para otras versiones

 XAMPP para Windows
7.2.2 (PHP 7.2.2)

 XAMPP para Linux
7.2.2 (PHP 7.2.2)

 XAMPP para OS X
XAMPP-VM (PHP 7.2.2)

New XAMPP-VM for OS X available!
[Try it now](#)

Ilustración 4 Página principal de XAMPP

Estando dentro página principal, debe seleccionar la opción que corresponda a su sistema operativo y empezará la descarga automáticamente.

Una vez haya finalizado la descarga de XAMPP, debe ejecutar el programa en modo administrador y aparecerá la siguiente ventana;

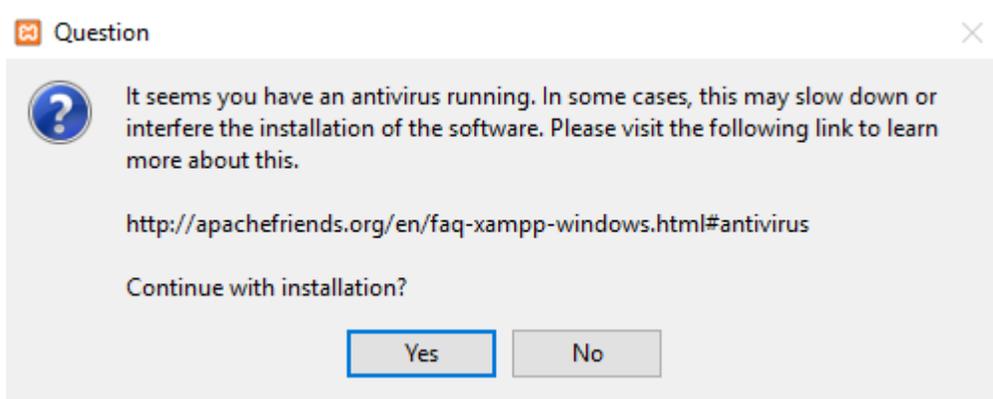


Ilustración 5 Recomendaciones antes de la instalación de XAMPP

Es una ventana informativa, la cual explica que para evitar algún inconveniente se debe pausar el antivirus que corresponda a su ordenador, al realizar la recomendación anterior se procede a pulsar el botón “Yes”. Una vez realizado este proceso aparecerá la siguiente ventana;

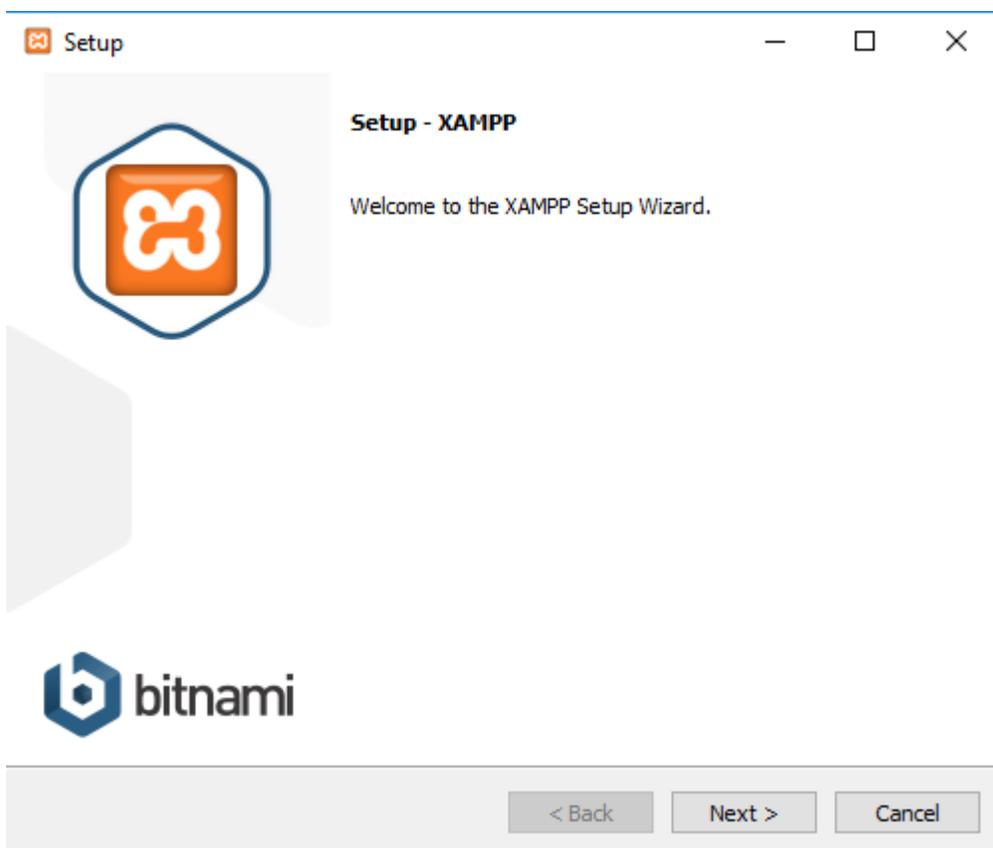


Ilustración 6 Proceso de la instalación de XAMPP

En la cual debe dar clic en el botón “Next” y sin modificar ninguna opción debe repetir el proceso hasta que la instalación finalice, al finalizar la instalación el programa quedará disponible en nuestro ordenador.

NODE.JS

“Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema más grande de librerías de código abierto en el mundo”.(Node.js, n.d.)

Para realizar su correcta instalación se debe dirigir a la página principal por medio del siguiente enlace; <https://nodejs.org/es/>



Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema mas grande de librerías de código abierto en el mundo.

Spectre and Meltdown in the context of Node.js.

Descargar para Windows (x64)

8.9.4 LTS
Recomendado para la mayoría

9.7.1 Actual
Últimas características

[Otras Descargas](#) | [Cambios](#) | [Documentación del API](#) [Otras Descargas](#) | [Cambios](#) | [Documentación del API](#)

Ó revise la [Agenda de LTS](#).

Ilustración 7 Página principal de node.js

En donde seleccionará la opción “8.9.4 LTS (recomendado para la mayoría)” e iniciará la descarga del programa automáticamente. Una vez finalizada la descarga del programa, debe ejecutarlo como administrador y aparecerá la siguiente ventana;

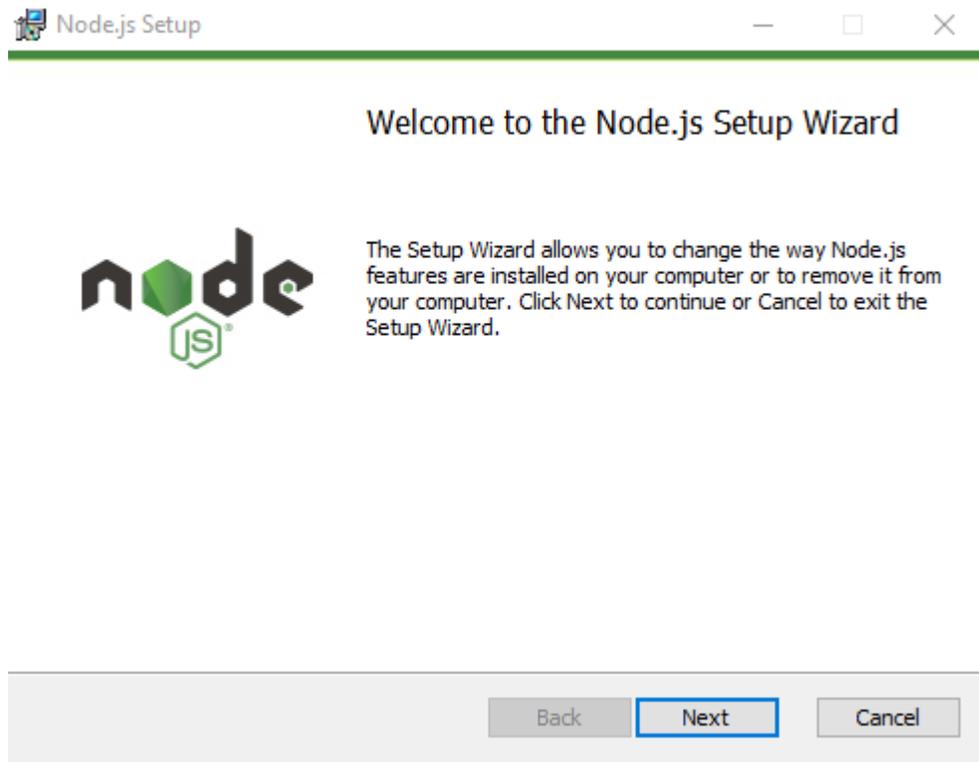


Ilustración 8 Proceso de instalación de Node.js

En donde dará clic en el botón “Next” y sin modificar ninguna opción procederá a realizar el mismo proceso hasta finalizar la instalación, de esta forma Node.js quedará instalado en nuestro ordenador.

GOOGLE CHROME

Google Chrome es un navegador diseñado por Google, el cual es requerido en este proyecto, porque Calisoft cuenta con una extensión desarrollada bajo este navegador, dicha extensión es usada en el módulo de Testing para generar las pruebas automatizadas, por esta razón es indispensable para su correcto funcionamiento.

El proceso de instalación de Google Chrome, consiste en dirigirse a la página de su descarga por medio del siguiente enlace; <https://www.google.com.mx/intl/es-419/chrome/#eula>

Obtén un navegador web gratuito y veloz

Un solo navegador para tu computadora, teléfono y tablet

Descargar Chrome

Para Windows 10/8.1/8/7 64-bit

Descargar Chrome para otra plataforma

Ilustración 9 Página de descarga de Google Chrome

Dentro de la página se debe pulsar el botón “Descargar Chrome” y automáticamente iniciará su descarga. Una vez descargado el programa, se debe iniciar como administrador. El proceso no requiere de alguna decisión por el usuario, sólo se ejecuta el programa y su instalación es automática.

INSTALACIÓN

Uso de XAMPP para ubicar la carpeta del proyecto

Una vez se hayan instalado los anteriores programas, se procederá a realizar la instalación del proyecto en el ordenador. El proyecto se puede obtener vía online en el siguiente repositorio Github; <https://github.com/Calisoft2017/ModuloCentral>. Ya con el proyecto en el ordenador, se procederá a almacenar en la carpeta Htdocs de Xampp, esta carpeta se encuentra en la siguiente ruta; C:\xampp\htdocs

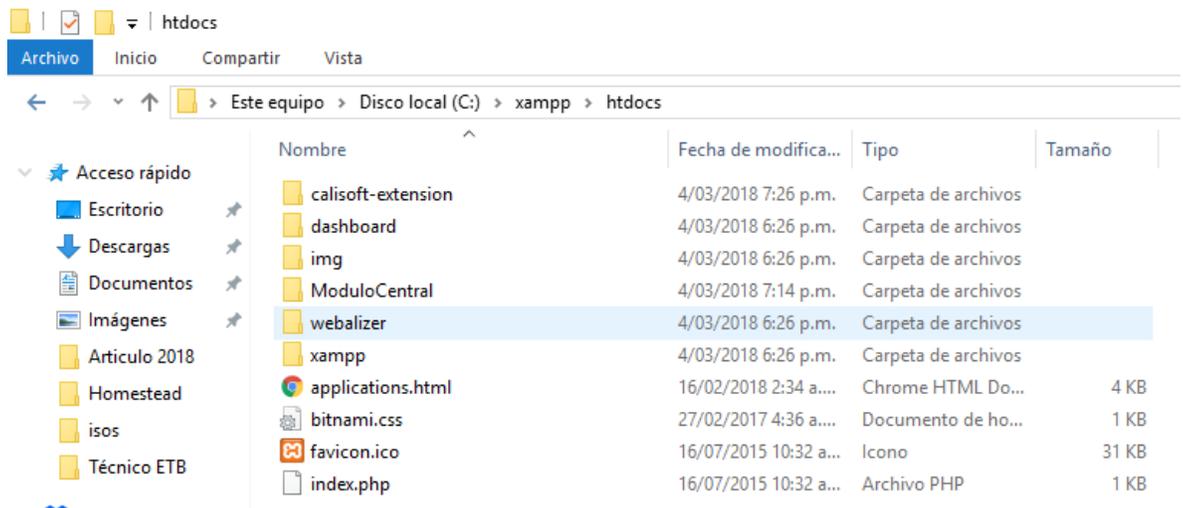


Ilustración 10 Ruta de la carpeta principal de Calisoft

En este caso, “ModuloCentral” es la carpeta del proyecto Calisoft. Una vez ubicada la carpeta procedemos a ejecutar la aplicación Xampp como administrador y pulsamos en el botón “Start” en la casilla de Apache y MySQL, como se ve a continuación;

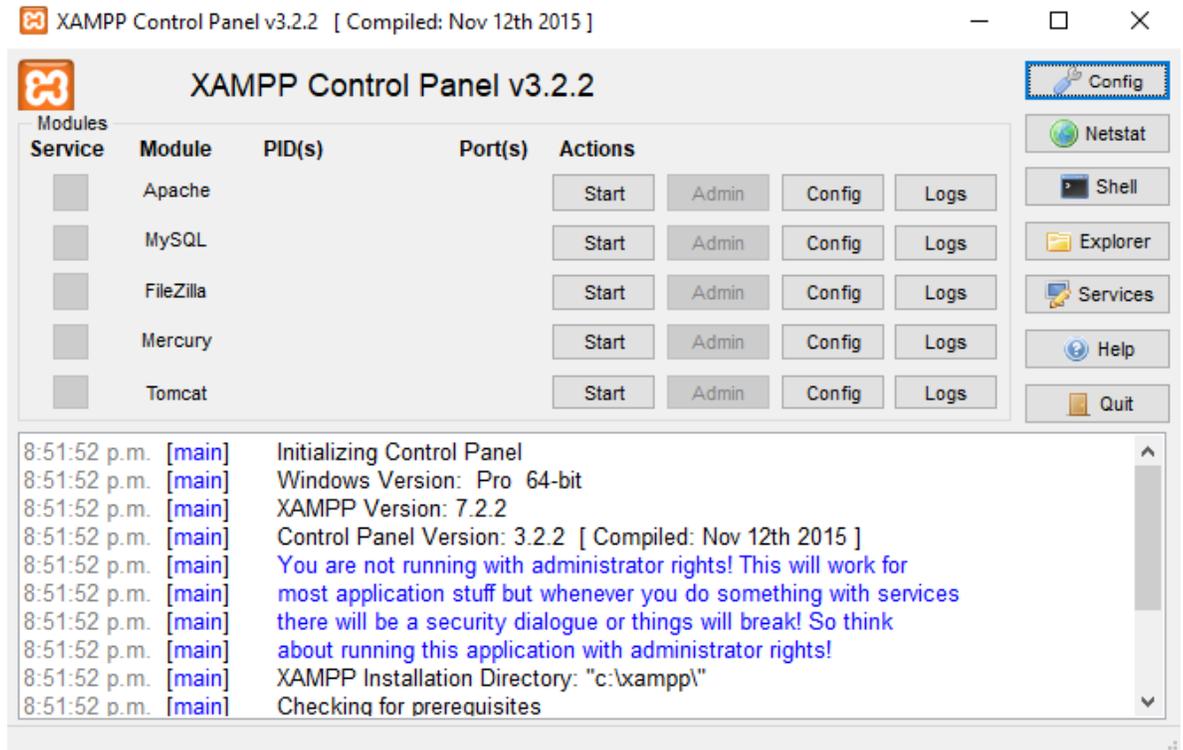


Ilustración 11 Interfaz XAMPP

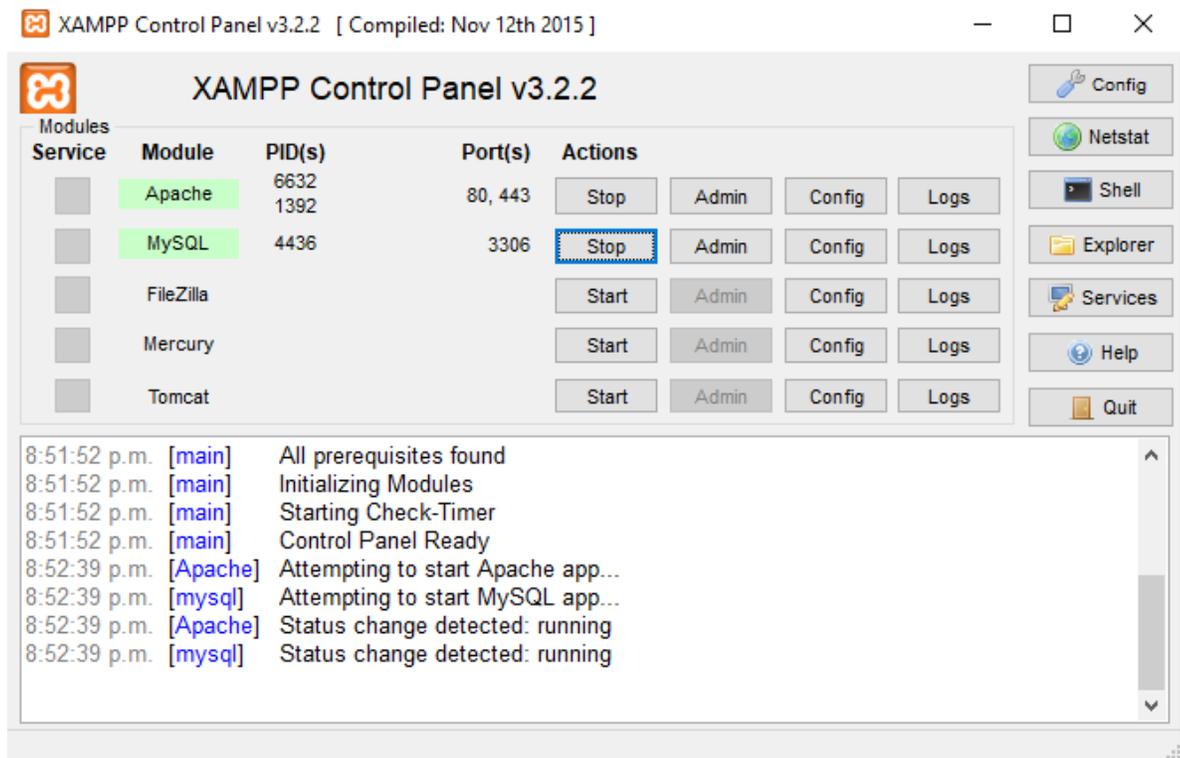


Ilustración 12 Interfaz XAMPP con servicios activados

Las casillas Apache y MySql deben quedar en color verde, lo que significa que el procedimiento fue exitoso, esto se hace para habilitar un servidor local y encender la base de datos que usará Calisoft.

Instalación de Composer en la carpeta del proyecto

Ahora, es necesario abrir un símbolo del sistema, para ello realizamos el siguiente comando; “Windows” + “R”

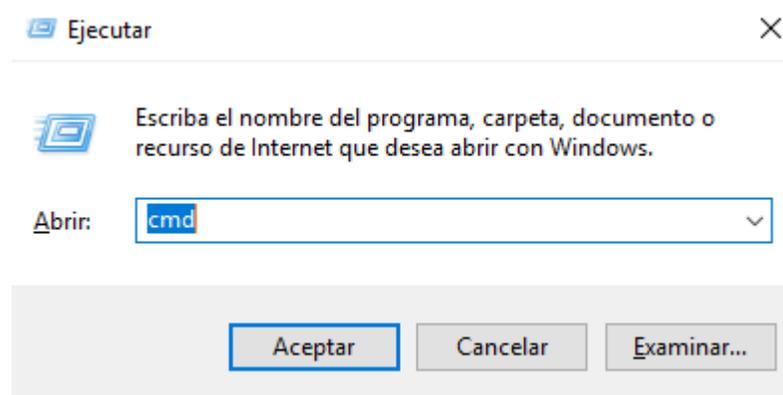


Ilustración 13 Ventana ejecutar

Aparecerá la ventana de “Ejecutar”, digitamos el comando “cmd” y damos clic en el botón “Aceptar”.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.16299.125]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Odin>
```

Ilustración 14 Símbolo del sistema

Se nos abrirá el símbolo del sistema ubicado en la carpeta del usuario del ordenador, en este caso el usuario principal es “Odin”, por medio del comando “cd..” retrocedemos al disco local “C”.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.16299.125]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Odin>cd..
```

Ilustración 15 Ejecución del comando “cd..”

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.16299.125]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Odin>cd..

C:\Users>cd..

C:\>
```

Ilustración 16 Ejecución del comando “cd..”

Una vez ubicados en el disco local “C” procedemos a buscar por medio del comando “cd nombreDeLaCarpeta” la carpeta en donde se alojó el proyecto Calisoft, la ruta en este caso es la siguiente;

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.16299.125]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Odin>cd..

C:\Users>cd..

C:\>cd xampp

C:\xampp>cd htdocs

C:\xampp\htdocs>cd modulocentral

C:\xampp\htdocs\ModuloCentral>
```

Ilustración 17 Cmd ubicación de la carpeta principal

Una vez estemos ubicados en la carpeta del proyecto, procederemos a ejecutar el comando “composer install” y pulsamos la tecla enter.

```
C:\xampp>cd htdocs

C:\xampp\htdocs>cd modulocentral

C:\xampp\htdocs\ModuloCentral>composer install
```

Ilustración 18 Composer install en CMD

```
C:\xampp\htdocs\ModuloCentral>composer install
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Nothing to install or update
Generating optimized autoload files
```

Ilustración 19 Procedimiento de la instalación de Composer en carpeta principal

Este comando procederá a ejecutar la instalación de Composer que procederá a instalar automáticamente en nuestra carpeta, esto es indispensable para el correcto funcionamiento de Calisoft.


```

C:\Windows\System32\cmd.exe
league/flysystem suggests installing league/flysystem-aws-s3-v3 (Allows you to use S3 storage with AWS SDK v3)
league/flysystem suggests installing league/flysystem-azure (Allows you to use Windows Azure Blob storage)
league/flysystem suggests installing league/flysystem-cached-adapter (Flysystem adapter decorator for metadata caching)
league/flysystem suggests installing league/flysystem-eventable-filesystem (Allows you to use EventableFilesystem)
league/flysystem suggests installing league/flysystem-rackspace (Allows you to use Rackspace Cloud Files)
league/flysystem suggests installing league/flysystem-sftp (Allows you to use SFTP server storage via phpseclib)
league/flysystem suggests installing league/flysystem-webdav (Allows you to use WebDAV storage)
league/flysystem suggests installing league/flysystem-ftp (Allows you to use FTP storage)
league/flysystem suggests installing spatie/flysystem-dropbox (Allows you to use Dropbox storage)
league/flysystem suggests installing srmklive/flysystem-dropbox-v2 (Allows you to use Dropbox storage for PHP 5 applications)
laravel/framework suggests installing aws/aws-sdk-php (Required to use the SQS queue driver and SES mail driver (~3.0).)
laravel/framework suggests installing doctrine/dbal (Required to rename columns and drop SQLite columns (~2.5).)
laravel/framework suggests installing ext-pcntl (Required to use all features of the queue worker.)
laravel/framework suggests installing ext-posix (Required to use all features of the queue worker.)
laravel/framework suggests installing guzzlehttp/guzzle (Required to use the Mailgun and Mandrill mail drivers and the ping methods on schedules (~6.0).)
laravel/framework suggests installing league/flysystem-aws-s3-v3 (Required to use the Flysystem S3 driver (~1.0).)
laravel/framework suggests installing league/flysystem-cached-adapter (Required to use Flysystem caching (~1.0).)
laravel/framework suggests installing league/flysystem-rackspace (Required to use the Flysystem Rackspace driver (~1.0).)
laravel/framework suggests installing nexmo/client (Required to use the Nexmo transport (~1.0).)
laravel/framework suggests installing pda/phenstalk (Required to use the beanstalk queue driver (~3.0).)
laravel/framework suggests installing symfony/dom-crawler (Required to use most of the crawler integration testing tools (~3.3).)
laravel/framework suggests installing symfony/psr-http-message-bridge (Required to psr7 bridging features (~1.0).)
psy/psysh suggests installing ext-pcntl (Enabling the PCNTL extension makes PsySH a lot happier :)
psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)
psy/psysh suggests installing ext-posix (If you have PCNTL, you'll want the POSIX extension as well.)
psy/psysh suggests installing hoa/console (A pure PHP readline implementation. You'll want this if your PHP install doesn't already support readline or libedit.)
predis/predis suggests installing ext-redis (Allows faster serialization and deserialization of the Redis protocol)
sebastian/global-state suggests installing ext-uopz (*)
phpunit/phpunit-mock-objects suggests installing ext-soap (*)
phpunit/php-code-coverage suggests installing ext-xdebug (~2.5.1)
phpunit/phpunit suggests installing ext-xdebug (*)
phpunit/phpunit suggests installing phpunit/php-invoker (~1.1)
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postInstall
> php artisan optimize

C:\xampp\htdocs\ModuloCentral>composer update
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 16 updates, 0 removals
 - Updating symfony/css-selector (v3.3.6 => v4.0.5): Loading from cache
 - Updating symfony/var-dumper (v3.3.6 => v3.4.5): Loading from cache
 - Updating symfony/routing (v3.3.6 => v3.4.5): Loading from cache
 - Updating symfony/process (v3.3.6 => v3.4.5): Loading from cache
 - Installing symfony/polyfill-php70 (v1.7.0): Loading from cache
 - Updating symfony/http-foundation (v3.3.6 => v3.4.5): Loading from cache
 - Updating symfony/debug (v3.3.6 => v3.4.5): Loading from cache
 - Updating symfony/event-dispatcher (v3.3.6 => v4.0.5): Loading from cache
 - Updating symfony/http-kernel (v3.3.6 => v3.4.5): Loading from cache
 - Updating symfony/finder (v3.3.6 => v3.4.5): Loading from cache
 - Updating symfony/console (v3.3.6 => v3.4.5): Loading from cache
 - Updating symfony/translation (v3.3.6 => v4.0.5): Loading from cache
 - Updating league/flysystem (1.0.42 => 1.0.43): Loading from cache
 - Updating doctrine/infllector (v1.2.0 => v1.3.0): Loading from cache
 - Updating laravel/framework (v5.5.35 => v5.5.36): Loading from cache
 - Updating symfony/yaml (v3.3.6 => v4.0.5): Loading from cache
 - Updating doctrine/instantiator (1.0.5 => 1.1.0): Loading from cache
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postUpdate
> php artisan optimize

C:\xampp\htdocs\ModuloCentral>

```

Ilustración 22 Actualización Composer en CMD

Creación de la base de datos

Una vez tengamos instalado y actualizado Composer en nuestra carpeta principal, procederemos a crear una base de datos, que se llamará DBS_Calisoft, en este caso se usó el programa Navicat, que sirve como gestor de base de datos, ustedes pueden usar el de su preferencia.

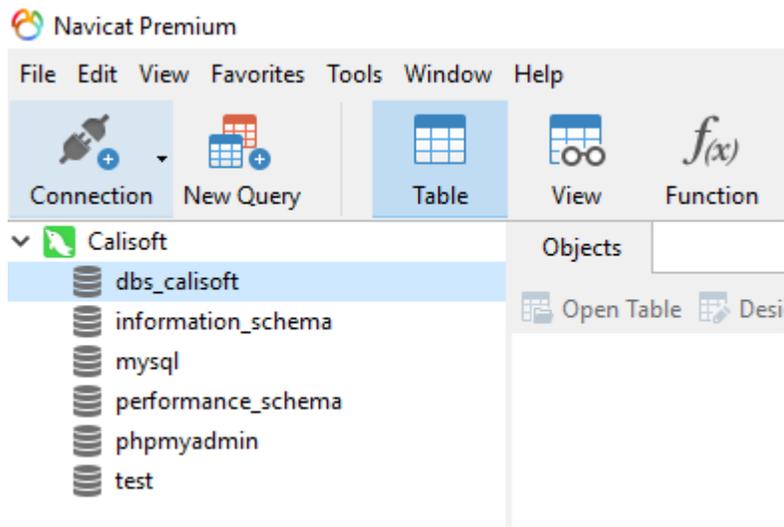


Ilustración 23 Creación base de datos

Tengan en cuenta el usuario y la contraseña con el que se crea la conexión, ya que posteriormente serán necesarios. En este paso sólo es necesario crear la base de datos.

Modificación del archivo `example.env`

Ahora, debemos ir a la carpeta principal del proyecto y buscamos el archivo “`example.env`”.

Este equipo > Disco local (C:) > xampp > htdocs > ModuloCentral

Nombre	Fecha de modifica...	Tipo	Tamaño
.git	4/03/2018 9:41 p.m.	Carpeta de archivos	
app	4/03/2018 9:41 p.m.	Carpeta de archivos	
bootstrap	4/03/2018 9:41 p.m.	Carpeta de archivos	
config	4/03/2018 9:41 p.m.	Carpeta de archivos	
database	4/03/2018 9:41 p.m.	Carpeta de archivos	
public	4/03/2018 9:41 p.m.	Carpeta de archivos	
resources	4/03/2018 9:41 p.m.	Carpeta de archivos	
routes	4/03/2018 9:41 p.m.	Carpeta de archivos	
storage	4/03/2018 9:41 p.m.	Carpeta de archivos	
tests	4/03/2018 9:41 p.m.	Carpeta de archivos	
vendor	4/03/2018 9:55 p.m.	Carpeta de archivos	
.gitattributes	4/03/2018 9:41 p.m.	Documento de tex...	1 KB
.gitignore	4/03/2018 9:41 p.m.	Documento de tex...	1 KB
artisan	4/03/2018 9:41 p.m.	Archivo	2 KB
composer.json	4/03/2018 9:41 p.m.	Archivo JSON	2 KB
composer.lock	4/03/2018 9:55 p.m.	Archivo LOCK	144 KB
example.env	4/03/2018 4:49 p.m.	Archivo ENV	1 KB
jsconfig.json	4/03/2018 9:41 p.m.	Archivo JSON	1 KB
package.json	4/03/2018 9:41 p.m.	Archivo JSON	2 KB
phpunit.xml	4/03/2018 9:41 p.m.	Documento XML	2 KB
readme.md	4/03/2018 9:41 p.m.	Archivo MD	2 KB
server.php	4/03/2018 9:41 p.m.	Archivo PHP	1 KB
webpack.mix.js	4/03/2018 9:41 p.m.	Archivo JavaScript	3 KB

Ilustración 24 Ubicación archivo "example.env"

Para modificar el archivo "example.env" es necesario abrirlo por medio de un editor de texto, en este caso usaremos el editor de texto Visual Studio Code para editar el archivo, usted puede usar el editor de texto de su preferencia.

```
example.env ●
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=
4 APP_DEBUG=true
5 APP_LOG_LEVEL=debug
6 APP_URL=http://localhost
7
8 DB_CONNECTION=mysql
9 DB_HOST=127.0.0.1
10 DB_PORT=3306
11 DB_DATABASE=dbs_calisoft
12 DB_USERNAME=root
13 DB_PASSWORD=
14
15 BROADCAST_DRIVER=log
16 CACHE_DRIVER=file
17 SESSION_DRIVER=file
18 QUEUE_DRIVER=database
19
20 REDIS_HOST=127.0.0.1
21 REDIS_PASSWORD=null
22 REDIS_PORT=6379
23
24 MAIL_DRIVER=smtp
25 MAIL_HOST=smtp.mailtrap.io
26 MAIL_PORT=2525
27 MAIL_USERNAME=26d4e5da1fb4e7
28 MAIL_PASSWORD=f82f27b88f9aeb
29 MAIL_ENCRYPTION=null
30
31 PUSHER_APP_ID=376363
32 PUSHER_APP_KEY=13fba520219a11216ae1
33 PUSHER_APP_SECRET=e6c109277502493e15e1
34 PUSHER_APP_CLUSTER=mt1
35
```

Ilustración 25 Contenido archivo "example.env"

Se debe modificar la sección "DB_DATABASE" con forme al nombre con el que se creó la base de datos, el username y el password deben corresponder a su configuración.

Configuración del repositorio de MAIL

Se debe configurar la sección de MAIL para que el aplicativo envíe mensajes vía correo, esta configuración debe realizarse de la siguiente manera;

```
MAIL_DRIVER=smtplib
MAIL_HOST=smtplib.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=26d4e5da1fb4e7
MAIL_PASSWORD=f82f27b88f9aeb
MAIL_ENCRYPTION=null
```

Ilustración 26 Configuración de MAIL

Es necesario que se dirija a la página <https://mailtrap.io/> y hacer uso de sus servicios, “Mailtrap es un servidor SMTP falso para que los equipos de desarrollo prueben, vean y compartan los correos electrónicos enviados desde los entornos de desarrollo y puesta en escena sin enviar spam a clientes reales.” (mailtrap, n.d.)

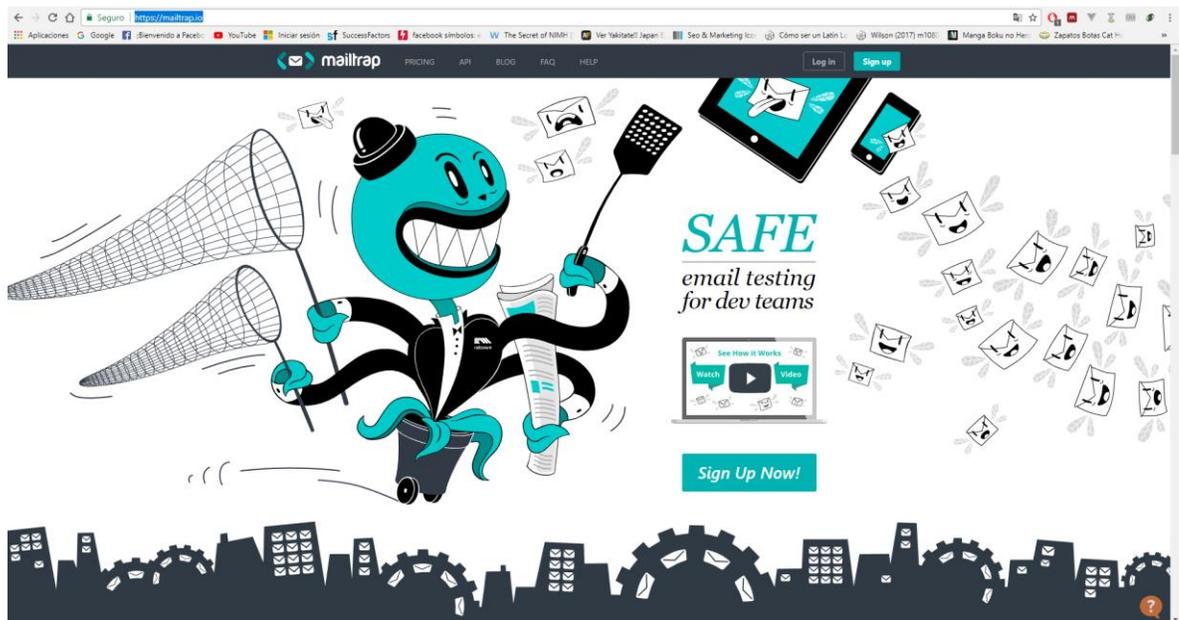


Ilustración 27 Página principal de Mailtrap

Una vez estando en la página principal, procederemos a dar clic en el botón “Sing up” el cual nos redirigirá a la siguiente ventana;

Create Free Forever Account

 Use Google account
 Use Github account
 ----- or -----
Signup with Your Email

 No credit card needed

 No Contract needed

Ilustración 28 Inicio de sesión Mailtrap

Donde pulsaremos en cualquiera de las tres opciones para crear una cuenta. Una vez cree el perfil e inicie sesión en la página, nos saldrá una ventana como la siguiente;


Shared Inboxes Billing
Upgrade
 Johan Suarez ▾

Home
New Company

My Inboxes   

Create Inbox

Inboxes	Messages	Max size	Last message	Action
 Demo inbox	✉ 42 / 50	50	3 hours ago	  

Ilustración 29 Backend Mailtrap

Por defecto se crea el repositorio de mensajes llamado; “Demo inbox”, daremos clic en el para ingresar a sus ajustes.

The screenshot shows the Mailtrap web interface. On the left is an email inbox with the following messages:

- Se te ha asignado un evaluador** (4 hours ago)
- Se te ha asignado un proyecto** (4 hours ago)
- Su proyecto fue aceptado ¡Felicitaciones!** (4 hours ago)
- Se ha enviado un formulario de un caso prueba** (7 days ago)
- Se ha creado un caso prueba para su proyecto** (7 days ago)
- Proyecto listo para evaluacion** (7 days ago)
- Se ha activado su proyecto** (8 days ago)
- Se ha enviado un formulario de un caso prueba** (8 days ago)
- Se ha creado un caso prueba para su proyecto** (8 days ago)
- Se ha enviado un formulario de un caso prueba** (8 days ago)
- Se ha creado un caso prueba para su proyecto** (8 days ago)
- Se ha enviado un formulario de un caso prueba** (8 days ago)
- Se ha creado un caso prueba para su proyecto** (8 days ago)
- Se ha enviado un formulario de un caso prueba** (8 days ago)

On the right, the SMTP Settings section is visible, including:

- Credentials** (Reset SMTP/POP3)
- SMTP**
 - Host: smtp.mailtrap.io
 - Port: 25 or 465 or 2525
 - Username: 26d4e5da1fb4e7
 - Password: f82f27b88f9aeb
 - Auth: PLAIN, LOGIN and CRAM-MD5
 - TLS: Optional
- Integrations**
 - Ruby on Rails (selected)

Below the integrations, there is a code block for ActionMailer configuration:

```

config.action_mailer.delivery_method = :smtp
config.action_mailer.smtp_settings = {
  :user_name => '26d4e5da1fb4e7',
  :password => 'f82f27b88f9aeb',
  :address => 'smtp.mailtrap.io',
  :domain => 'smtp.mailtrap.io',
  :port => '2525',
  :authentication => :cram_md5
}

```

Ilustración 30 Administración de mensajes de Mailtrap

Una vez dentro, iremos a la sección “Integrations” y en el input selec llamado; “Ruby on Rails” buscaremos y seleccionaremos la opción de “Laravel”

Auth: PLAIN, LOGIN and CRAM-MD5
TLS: Optional

Integrations

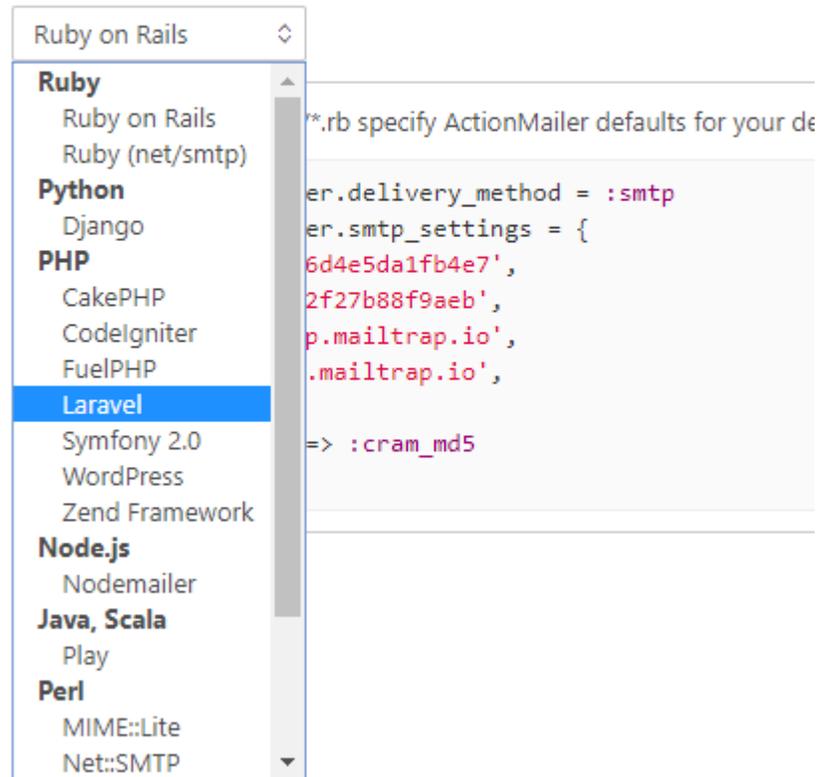


Ilustración 31 Apartado Integrations Mailtrap

En la parte de abajo se actualizarán los datos necesarios para configurar mailtrap en Laravel

```
return array(  
    "driver" => "smtp",  
    "host" => "smtp.mailtrap.io",  
    "port" => 2525,  
    "from" => array(  
        "address" => "from@example.com",  
        "name" => "Example"  
    ),  
    "username" => "26d4e5da1fb4e7",  
    "password" => "f82f27b88f9aeb",  
    "sendmail" => "/usr/sbin/sendmail -bs",  
    "pretend" => false  
);
```

Ilustración 32 Atributos de Mailtrap en Laravel

Estos datos debes ser usados en el archivo `example.env` en la sección de MAIL. Al realizar las correspondientes modificaciones, debe quedar de la siguiente forma;

```
MAIL_DRIVER=smtp  
MAIL_HOST=smtp.mailtrap.io  
MAIL_PORT=2525  
MAIL_USERNAME=26d4e5da1fb4e7  
MAIL_PASSWORD=f82f27b88f9aeb  
MAIL_ENCRYPTION=null
```

Ilustración 33 Apartado de Mail en archivo example.env

Configuración de Pusher para notificaciones en tiempo real

Ahora, procederemos a editar la sección de PUSHER, que es la necesaria para que las notificaciones lleguen en tiempo real, para ello iremos a la página <https://pusher.com/>

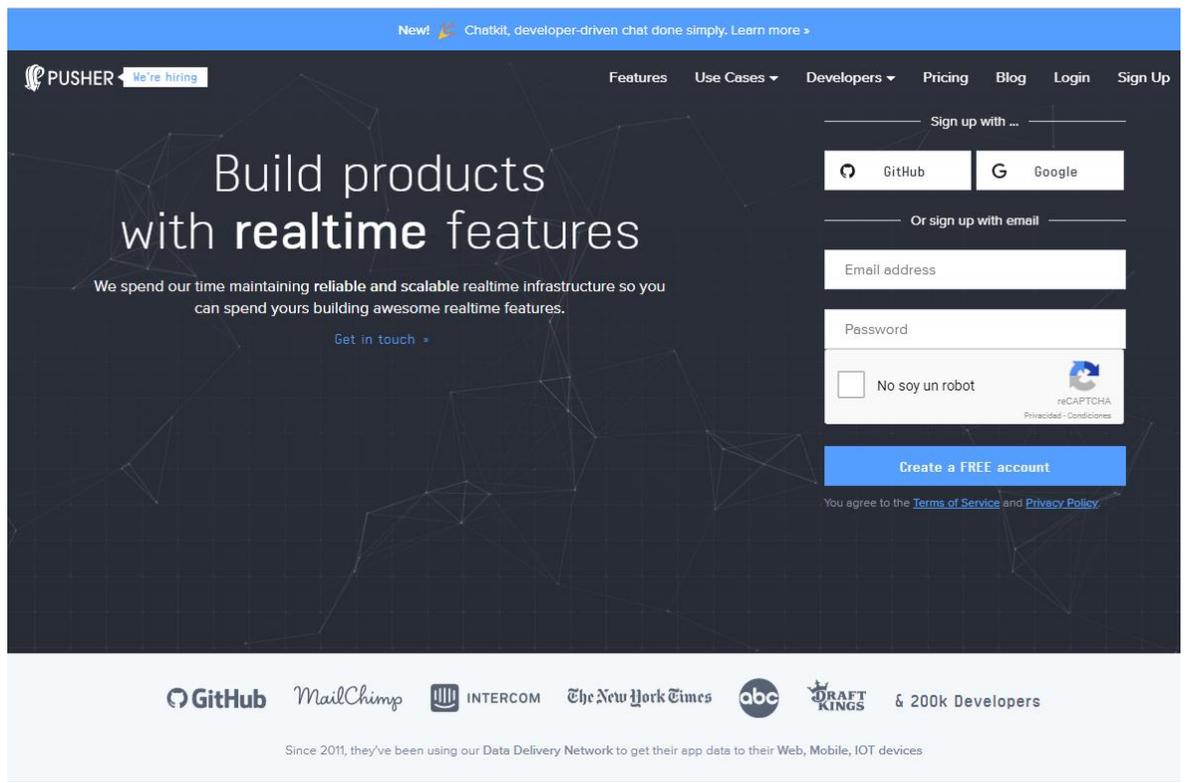


Ilustración 34 Página principal de Pusher

Para usar este servicio es necesario crear una cuenta, esto se puede hacer por medio de un usuario en Github, Google o digitando un correo y aportando una contraseña. Una vez creada la cuenta e iniciada la sesión, la primera pestaña en aparecer será la siguiente;

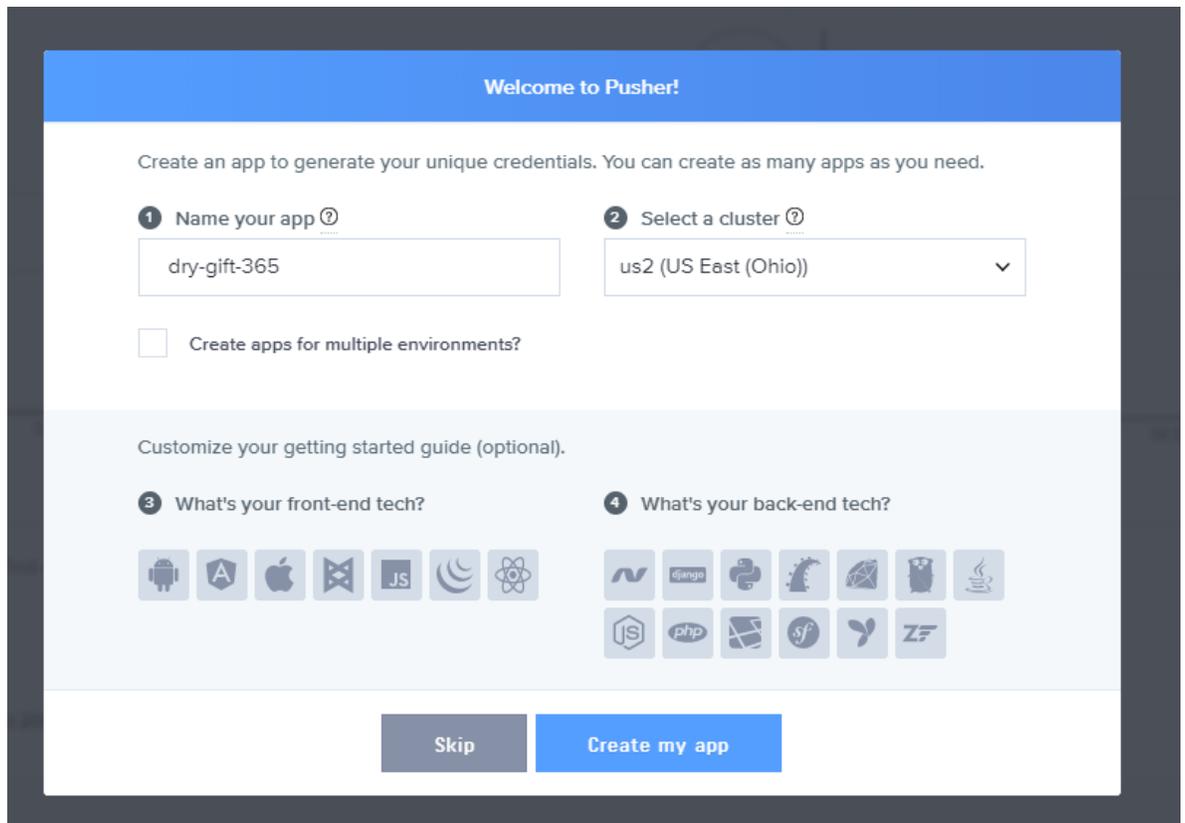


Ilustración 35 Welcome to Pusher

En donde se debe modificar el nombre de la “App” a preferencia del usuario y es necesario seleccionar el icono de Laravel como se ve en la siguiente imagen.

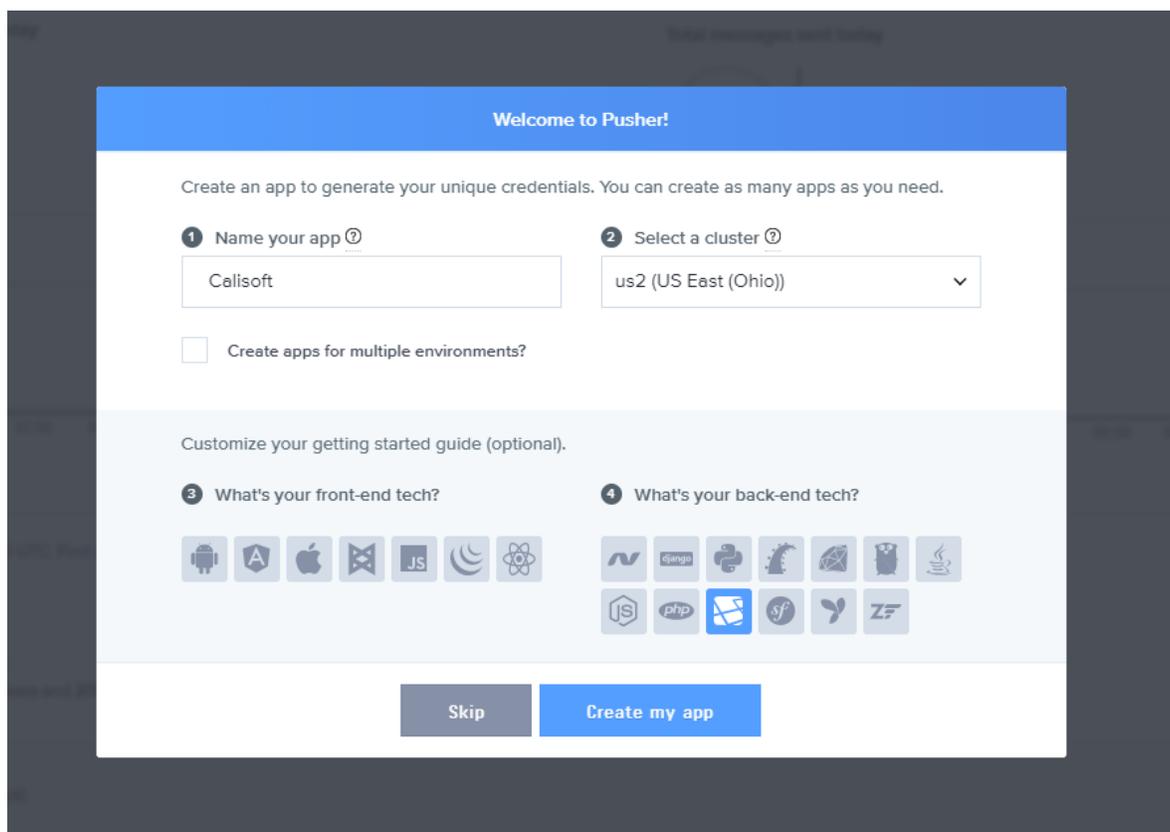


Ilustración 36 Modificación de Pusher

Para continuar se debe oprimir el botón “Create my app” y la página redirigirá a un Dash board en donde contiene todo el contenido y la administración de nuestra herramienta de tiempo real. Para entrelazar Pusher con Laravel, es necesario que se dirija a la pestaña “App Keys”

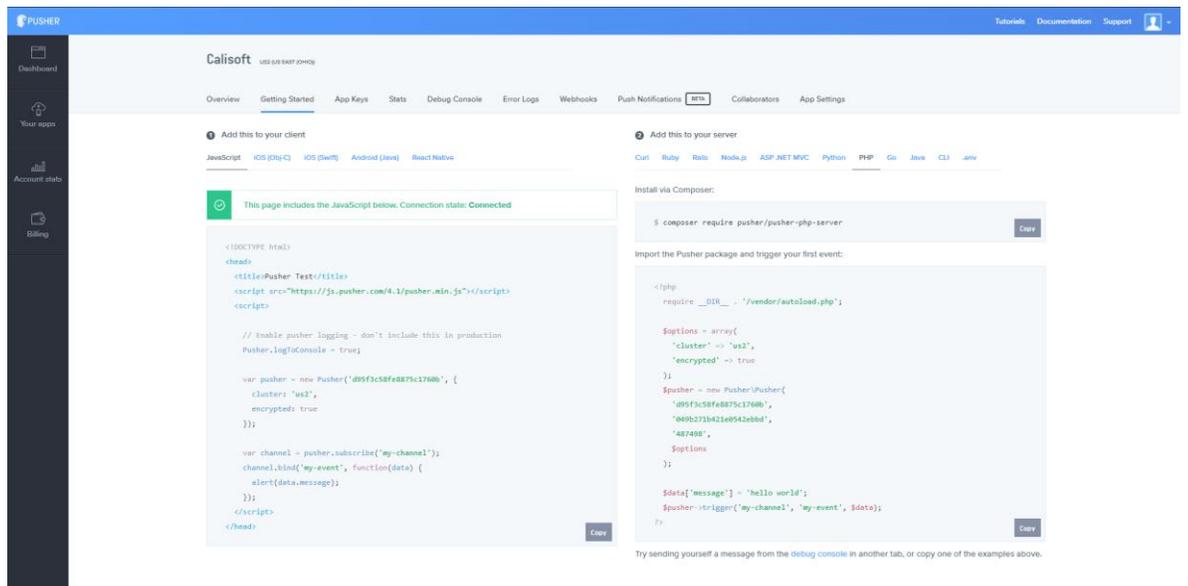


Ilustración 37 Backend Pusher

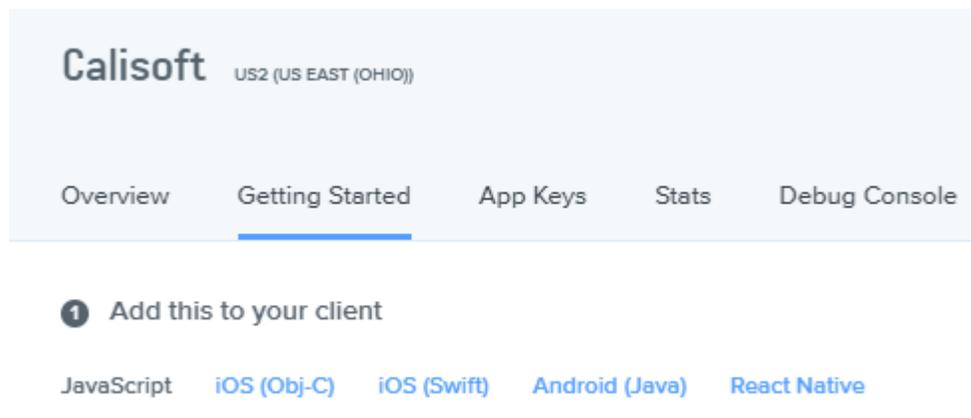


Ilustración 38 Pestañas Pusher

Seleccionamos la pestaña “App Keys” y una vez dentro, el sistema nos arrojará las diferentes llaves necesarias para que Calisoft conecte satisfactoriamente con Pusher

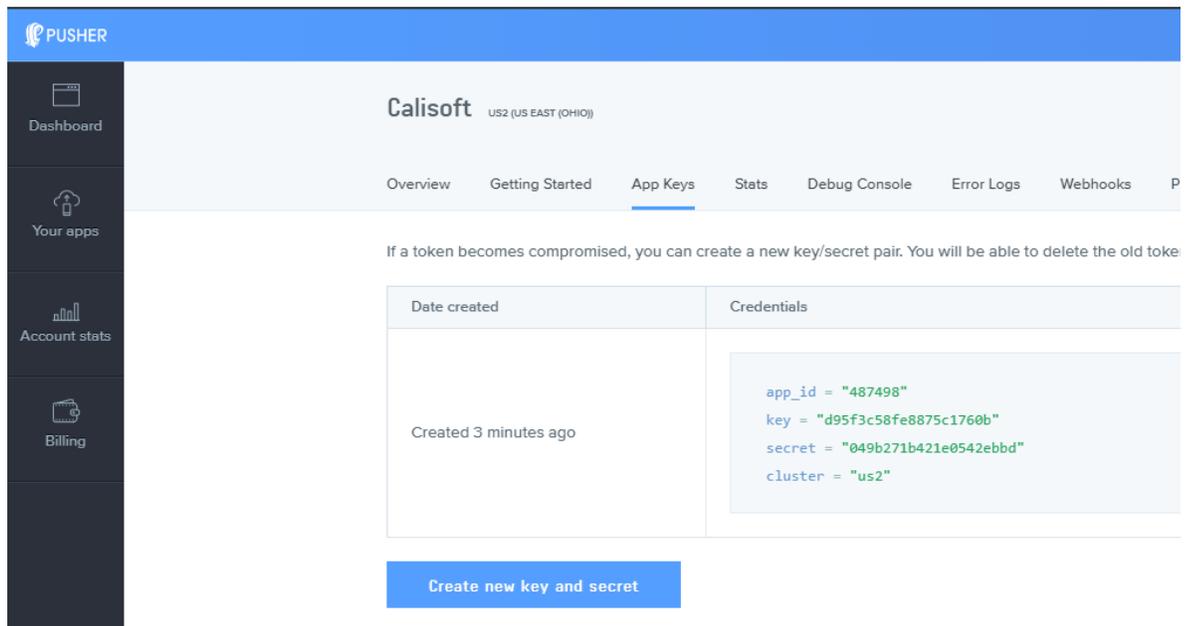


Ilustración 39 App Keys Pusher

Teniendo en cuenta las llaves, se procede a dejar la sección de Pusher del archivo `example.env`, de la misma forma en la que se halla en la página.

```
PUSHER_APP_ID=376363
PUSHER_APP_KEY=13fba520219a11216ae1
PUSHER_APP_SECRET=e6c109277502493e15e1
PUSHER_APP_CLUSTER=mt1
```

Ilustración 40 Configuración Pusher en archivo `example.env`

Cambiar nombre archivo `example.env`

Con el anterior proceso finalizamos la edición del archivo `example.env`, ahora, procedemos a modificar el nombre de dicho archivo, se recomienda realizar este proceso desde un editor de texto. Para este ejemplo se usó Visual Studio Code, dando clic derecho sobre el archivo y yendo a la opción de modificar nombre.

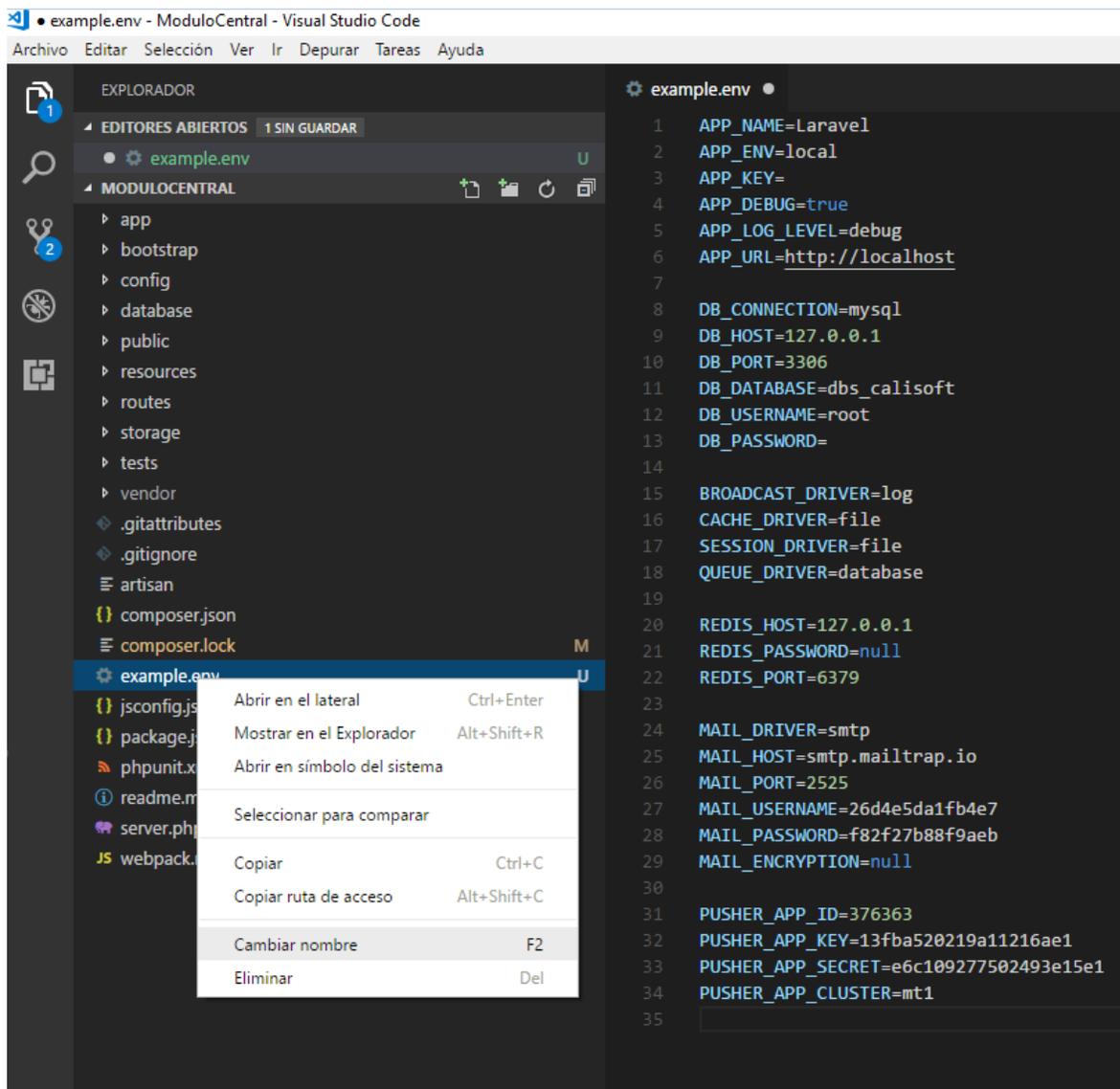


Ilustración 41 Cambiar nombre del archivo example.env

El archivo debe quedar como “.env”, sólo con la extensión como se aprecia en la siguiente ilustración;

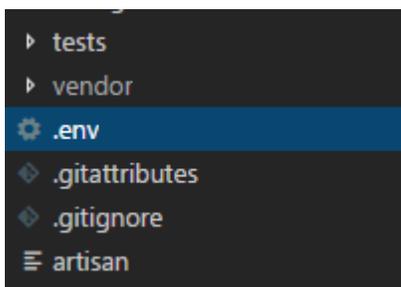


Ilustración 42 Archivo .env

Llave de acceso

Al finalizar este paso, procedemos a ajustar la llave de acceso del aplicativo, y para ello, usamos en el símbolo del sistema que anteriormente teníamos abierto, o simplemente abrimos uno nuevo como se explicó anteriormente y digitamos el código “php artisan key:generate”, como se ve en la siguiente ilustración;

```
C:\>cd xampp
C:\xampp>cd htdocs
C:\xampp\htdocs>cd modulocentral
C:\xampp\htdocs\ModuloCentral>php artisan key:generate
```

Ilustración 43 Php artisan key:generate

Presionamos la tecla enter y automáticamente asignará una Key para nuestro aplicativo.

Creación ruta a carpeta Storage

A Continuación procedemos a digitar el comando “php artisan storage:link” con el objetivo de crear una carpeta storage de acceso directo dentro del contenido public, esto se hace para generar seguridad al momento de realizar una lectura a los archivos almacenados en la nube. Presionamos la tecla enter y si el proceso es satisfactorio deberá aparecer de la siguiente forma como se aprecia en la siguiente ilustración;

```
C:\xampp\htdocs\ModuloCentral>php artisan storage:link
The [public/storage] directory has been linked.
```

Ilustración 44 Proceso exitoso del comando Php artisan storage:link

Si al ejecutar el comando “php artisan storage:link” generar un error, se aconseja abrir el símbolo del sistema como administrador y volver a intentar.

Instalación de Vue de Node.js en la carpeta del proyecto

Al finalizar el paso anterior, se procede a instalar Vue de Node.Js en el aplicativo, lo cual se realiza digitando el siguiente comando en el símbolo del sistema; “npm install --only=dev”, este proceso tarda entre alrededor de 5 minutos. Una vez finalizado, debe mostrar una ventana como la siguiente;

```

C:\xampp\htdocs\ModuloCentral>npm install --only-dev
npm WARN deprecated gulp-util@3.0.8: gulp-util is deprecated - replace it, following the guidelines at https://medium.com/gulpjs/gulp-util-ca3b1f9f9ac5
> node-sass@4.7.2 install C:\xampp\htdocs\ModuloCentral\node_modules\node-sass
> node scripts/install.js

Cached binary found at C:\Users\Odin\AppData\Roaming\npm-cache\node-sass\4.7.2\win32-x64-57_binding.node
> uglifyjs-webpack-plugin@0.4.6 postinstall C:\xampp\htdocs\ModuloCentral\node_modules\webpack\node_modules\uglifyjs-webpack-plugin
> node lib/post_install.js

> node-sass@4.7.2 postinstall C:\xampp\htdocs\ModuloCentral\node_modules\node-sass
> node scripts/build.js

Binary found at C:\xampp\htdocs\ModuloCentral\node_modules\node-sass\vendor\win32-x64-57\binding.node
Testing binary
Binary is fine

> gifsicle@3.0.4 postinstall C:\xampp\htdocs\ModuloCentral\node_modules\gifsicle
> node lib/install.js

  ✓ gifsicle pre-build test passed successfully

> mozjpeg@5.0.0 postinstall C:\xampp\htdocs\ModuloCentral\node_modules\mozjpeg
> node lib/install.js

  ✓ mozjpeg pre-build test passed successfully

> optipng-bin@3.1.4 postinstall C:\xampp\htdocs\ModuloCentral\node_modules\optipng-bin
> node lib/install.js

  ✓ optipng pre-build test passed successfully

> pngquant-bin@3.1.1 postinstall C:\xampp\htdocs\ModuloCentral\node_modules\pngquant-bin
> node lib/install.js

  ✓ pngquant pre-build test passed successfully

> undefined postinstall C:\xampp\htdocs\ModuloCentral
> npm run prod

> @ prod C:\xampp\htdocs\ModuloCentral
> npm run production

> @ production C:\xampp\htdocs\ModuloCentral
> cross-env NODE_ENV=production node_modules\webpack\bin\webpack.js --progress --hide-modules --config=node_modules/laravel-mix/setup/webpack.config.js
91% additional asset processing

```

Ilustración 45 Proceso de instalación de Vue

```

C:\WINDOWS\system32\cmd.exe
DONE Compiled successfully in 147842ms

  Asset      Size  Chunks             Chunk Names
  \js\evaluator-proyectos.js 261 kB    14 [emitted] [big] \js\evaluator-proyectos
  \js\evaluator-script.js 1.17 MB    0 [emitted] [big] \js\evaluator-script
  \js\archivo-sql.js 1.13 MB    2 [emitted] [big] \js\archivo-sql
  \js\scripts-codificacion.js 1.13 MB    3 [emitted] [big] \js\scripts-codificacion
  \js\proyectos.js 269 kB    4 [emitted] [big] \js\proyectos
  \js\admin-proyectos.js 267 kB    5 [emitted] [big] \js\admin-proyectos
  \js\plataforma-student.js 330 kB    6 [emitted] [big] \js\plataforma-student
  \js\escenario.js 411 kB    7 [emitted] [big] \js\escenario
  \js\componentes.js 346 kB    8 [emitted] [big] \js\componentes
  \js\plataforma.js 257 kB    9 [emitted] [big] \js\plataforma
  \js\usuarios.js 407 kB   10 [emitted] [big] \js\usuarios
  \js\modelacion.js 258 kB   11 [emitted] [big] \js\modelacion
  \js\evaluator-docs.js 265 kB   12 [emitted] [big] \js\evaluator-docs
  \js\documentos.js 260 kB   13 [emitted] [big] \js\documentos
  \js\base-datos.js 1.22 MB    1 [emitted] [big] \js\base-datos
  \js\categorias.js 256 kB   15 [emitted] [big] \js\categorias
  \js\evaluacion-modelado.js 263 kB   16 [emitted] [big] \js\evaluacion-modelado
  \js\codificacion.js 255 kB   17 [emitted] [big] \js\codificacion
  \js\tipo-nomenclatura.js 192 kB   18 [emitted] \js\tipo-nomenclatura
  \js\nomenclaturas-show.js 253 kB   19 [emitted] [big] \js\nomenclaturas-show
  \js\items-codificacion.js 192 kB   20 [emitted] \js\items-codificacion
  \js\categorias-show.js 254 kB   21 [emitted] [big] \js\categorias-show
  \js\tipo-documento.js 192 kB   22 [emitted] \js\tipo-documento
  \js\semilleros.js 193 kB   23 [emitted] \js\semilleros
  \js\items-show.js 190 kB   24 [emitted] \js\items-show
  \js\notificaciones.js 189 kB   25 [emitted] \js\notificaciones
  \js\invitaciones.js 189 kB   26 [emitted] \js\invitaciones
  \js\bootstrap.js 189 kB   27 [emitted] \js\bootstrap

npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN uiv@0.11.11 requires a peer of bootstrap@^3.3.7 but none is installed. You must install or update bootstrap
npm WARN ajv-keywords@3.1.0 requires a peer of ajv@^6.0.0 but none is installed. You must install or update ajv
npm WARN @vue/test-utils@1.0.0-beta.12 requires a peer of vue-server-renderer@2.x but none is installed. You must install or update vue-server-renderer
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.3 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.3: wanted {"os": "darwin", "cpu": "x64", "libc": "glibc"}

added 1532 packages in 652.237s

C:\xampp\htdocs\ModuloCentral>

```

Ilustración 46 Finalización exitosa de Vue en carpeta del proyecto

Esto significa que todos los componentes se instalaron correctamente. Se recomienda que después de la instalación, se ejecute el comando “npm update” para que actualice los componentes instalados.

Migraciones

El paso siguiente se centra en generar los datos necesarios para agregar valores automáticamente a la base de datos, estos datos permitirán acceder a los diferentes módulos del aplicativo. El comando es “php artisan migrate”, se debe realizar sobre la carpeta del proyecto como se ha realizado con los comandos anteriores. (Recuerde que Apache y MySql deben estar activados en el programa Xampp).

```
C:\WINDOWS\system32\cmd.exe

C:\xampp\htdocs\ModuloCentral>php artisan migrate
```

Ilustración 47 Comando php artisan migrate

Se pulsa la tecla enter y si el proceso es satisfactorio, deberá aparecer la siguiente ventana;

```
Migrating: 2017_08_10_221755_create_item_codificacion_table
Migrated: 2017_08_10_221755_create_item_codificacion_table
Migrating: 2017_08_10_222153_create_scripts_table
Migrated: 2017_08_10_222153_create_scripts_table
Migrating: 2017_08_10_223052_create_nota_codificacion_table
Migrated: 2017_08_10_223052_create_nota_codificacion_table
Migrating: 2017_08_10_224221_create_modulos_table
Migrated: 2017_08_10_224221_create_modulos_table
Migrating: 2017_08_10_224416_create_comentarios_table
Migrated: 2017_08_10_224416_create_comentarios_table
Migrating: 2017_08_14_013303_create_tipo_bd_table
Migrated: 2017_08_14_013303_create_tipo_bd_table
Migrating: 2017_08_14_014415_create_type_nomenclatura_table
Migrated: 2017_08_14_014415_create_type_nomenclatura_table
Migrating: 2017_08_14_021004_create_archivo_bd_table
Migrated: 2017_08_14_021004_create_archivo_bd_table
Migrating: 2017_08_14_025019_create_calificacion_bd_table
Migrated: 2017_08_14_025019_create_calificacion_bd_table
Migrating: 2017_09_17_005624_create_jobs_table
Migrated: 2017_09_17_005624_create_jobs_table
Migrating: 2017_10_07_051306_create_inputs_type_table
Migrated: 2017_10_07_051306_create_inputs_type_table
Migrating: 2018_01_31_002018_create_test_values_table
Migrated: 2018_01_31_002018_create_test_values_table
Migrating: 2018_02_11_214036_create_items_evaluados_table
Migrated: 2018_02_11_214036_create_items_evaluados_table

C:\xampp\htdocs\ModuloCentral>
```

Ilustración 48 Migraciones exitosas

Para confirmar la acción de este código, puede dirigirse a la base de datos y comprobar la creación de dichas tablas.

Si durante el proceso se genera un error, verifique que el nombre, contraseña y usuario de la base de datos corresponda a los datos que se encuentran en el archivo “.env”.

Seeders

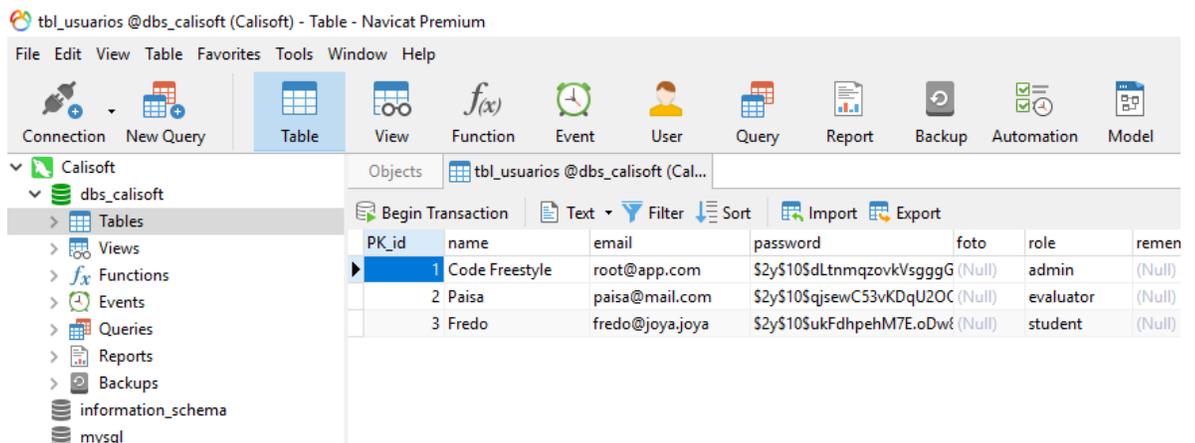
El paso siguiente al finalizar este proceso, es generar los seeders. Los seeders son un método de Laravel para generar registros automáticos en las tablas de la base de datos, esto es principal para definir los primeros usuarios del aplicativo o los registros principales con los que se quiere iniciar el software. El código es “php artisan db:seed”. Al dar enter se ejecutará el comando y al finalizar deberá aparecer la siguiente información;

```
C:\xampp\htdocs\ModuloCentral>php artisan db:seed
Seeding: UsersTableSeeder
Seeding: CategoriesTableSeeder
Seeding: TiposDocumentoTableSeeder
Seeding: SemillerosTableSeeder
Seeding: GrupoInvestigacionTableSeeder
Seeding: ItemsCodificacionTableSeeder
Seeding: ModulosTableSeeder
Seeding: TipoNomenclaturaTableSeeder
Seeding: TipoBdTableSeeder
Seeding: ComponenteTableSeeder
Seeding: TipoInputsTestingSeeder
Seeding: TestValueSeeder

C:\xampp\htdocs\ModuloCentral>
```

Ilustración 49 Comando php artisan db:seed exitoso

Para confirmar si los registros se insertaron correctamente, dirijase a la base de datos y compruebe si las tablas, por ejemplo, TBL_Usuarios contienen registros.



The screenshot shows the Navicat Premium interface for a MySQL database named 'db_calisoft'. The table 'tbl_usuarios' is selected, and its contents are displayed in a grid view. The table has 7 columns: PK_id, name, email, password, foto, role, and remen. There are 3 records in the table.

PK_id	name	email	password	foto	role	remen
1	Code Freestyle	root@app.com	\$2y\$10\$dLtnmqzovkVsgggG	(Null)	admin	(Null)
2	Paiza	paiza@mail.com	\$2y\$10\$qjsewC53vKDqU20C	(Null)	evaluator	(Null)
3	Fredo	fredo@joya.joya	\$2y\$10\$SukFdhpehM7E.oDw{	(Null)	student	(Null)

Ilustración 50 Registros de la tabla TBL_Usuarios

Como puede observar, el password se almacena con un Has, lo que no permite su lectura a simple vista, para saber las contraseñas de los diferentes usuarios, usted deberá dirigirse mediante su editor de texto a la siguiente carpeta siguiendo la ruta;

C:\xampp\htdocs\ModuloCentral\database\seeds\UsersTableSeeder.php y dentro del código usted podrá observar las correspondientes contraseñas de los usuarios creados mediante los métodos Seeders.

```
DB::table('TBL_Usuarios')->insert([
    [
        'name' => 'Code Freestyle',
        'email' => 'root@app.com',
        'role' => 'admin',
        'password' => bcrypt('12345')
    ],
    [
        'name' => 'Paisa',
        'email' => 'paisa@mail.com',
        'role' => 'evaluator',
        'password' => bcrypt('12345')
    ],
    [
        'name' => 'Fredo',
        'email' => 'fredo@joya.joya',
        'role' => 'student',
        'password' => bcrypt('12345')
    ]
])
```

Ilustración 51 Datos de los seeders

Correr el aplicativo

Hasta este punto usted ya logró configurar correctamente el aplicativo Calisoft, ahora usted deberá correr los siguientes códigos en diferentes símbolos del sistema, los códigos son;

- “php artisan serve”: Es necesario para crear acceder al servidor local mediante la dirección Ip que este comando genera.
- “npm run watch”: Este comando permite que los componentes .js se mantengan compilando.
- “php artisan queue:work --tries=3”: Es el encargado de realizar las notificaciones en tiempo real.

Recuerde que cada código debe ser ejecutado sobre la carpeta del aplicativo y en diferentes símbolos del sistema.

Acceso al login

Para acceder al Login, debe copiar la dirección Ip que generó el código “php artisan serve” Y pegarlo en el navegador de Google Chrome

```
C:\WINDOWS\system32\cmd.exe - php artisan serve

C:\xampp\htdocs\ModuloCentral>php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

Ilustración 52 Ip del servidor local

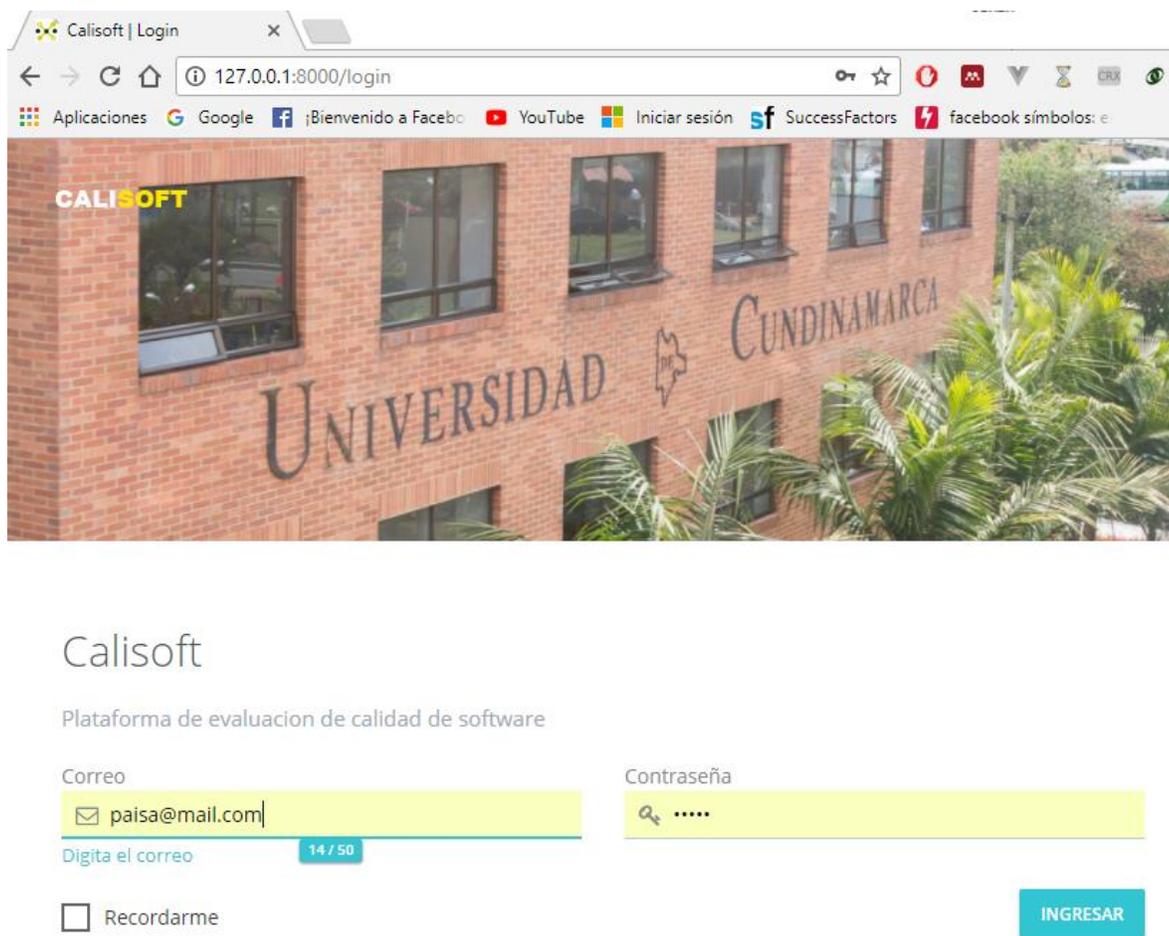


Ilustración 53 Página principal de Calisoft

Con este último paso procedemos a finalizar el proceso de instalación de Calisoft, para obtener información de los diferentes módulos del aplicativo, proceda a leer el Manual De Usuario de Calisoft.

REFERENCIAS

Apache Friends. (2015). About the XAMPP project. Retrieved March 4, 2018, from <https://www.apachefriends.org/de/about.html>

Composer. (n.d.). Introduction - Composer. Retrieved March 4, 2018, from <https://getcomposer.org/doc/00-intro.md>

mailtrap. (n.d.). mailtrap.io. Retrieved March 8, 2018, from <https://mailtrap.io/>

Node.js. (n.d.). Node.js. Retrieved March 4, 2018, from <https://nodejs.org/es/>

Anexo C. Articulo ponencia universidad san buenaventura.

Caracterización de estándares de codificación aplicando una métrica de evaluación para medir el nivel de cohesión en un software de calidad

Ing. Cesar Yesid Barahona Rodríguez, **Hector Hernan Castellanos, *John Fredy Osorio*

**Facultad de ingeniería, Universidad de Cundinamarca, Programa Ingeniería de sistemas
Facatativá, Cundinamarca, Colombia
cbarahona@ucundinamarca.edu.co*

***Facultad de ingeniería, Universidad de Cundinamarca, Programa Ingeniería de sistemas
Facatativá, Cundinamarca, Colombia
hhcastellanos@ucundinamarca.edu.co*

****Facultad de ingeniería, Universidad de Cundinamarca, Programa Ingeniería de sistemas
Facatativá, Cundinamarca, Colombia
jfosorio@ucundinamarca.edu.co*

Resumen

PHP Standards Recommendations (PSR) es un estándar de codificación que es ampliamente utilizado en la industria de software; al implementar un estándar como éste, permite obtener un código organizado y mejor desarrollado al cual se le pueda realizar mantenimiento de una forma sencilla y rápida, este estándar se puede caracterizar con el objetivo de encontrar directrices para evaluar el grado de cohesión entre el código implementado y la asimilación de estándares de codificación, con la finalidad de obtener un producto de software con características de calidad. Es por esto que se busca diseñar una métrica de evaluación que permita identificar cuáles de estas directrices representan una mayor importancia, con el objeto de generar una valoración de la aplicación del estándar.

Abstract

PHP standards Recommendations (PSR) is an encoding standard that is widely used in the software industry; When implementing a standard like this, Allows you to obtain an organized and better developed code that can be easily and quickly maintained, This standard can be characterized with the objective of finding guidelines to evaluate the degree of cohesion between the implemented code and the assimilation of coding standards, With the purpose of obtaining a

software product with quality characteristics. This is why it is sought to design an evaluation metric that allows identifying which of these guidelines represent a greater importance, in order to generate a valuation of the application of the standard.

1. Introducción

Hoy en día una de las áreas que presenta mayor crecimiento y constante cambio es la industria tecnológica, cada día nacen más empresas que deciden tomar como actividad económica el desarrollo de software o actividades relacionadas con el sector tecnológico, es por esto que buscan dar un valor agregado a sus productos y/o servicios, para lograr esto muchas de estas deciden utilizar estándares de calidad como lo es la norma ISO/IEC 25000 conocida como requisitos y evaluación de calidad de productos de software, una de estas características es la mantenibilidad que se puede explicar cómo la capacidad que tiene un sistema para poder realizar cambios sobre sus componentes de una manera efectiva.[1] Cabe resaltar que muchos de estos cambios no lo realiza la persona que creó el software en un principio; es por esto que las empresas deciden aplicar estándares de codificación en sus trabajos para que estos den la impresión que han sido hechos por un solo programador y puedan ser comprendidos en un futuro por otros programadores.

Por lo tanto, se propone desarrollar una herramienta que evalúe el estándar de codificación previamente establecido por la institución en los proyectos de desarrollo de software, además que esta pueda generar una calificación en base a una métrica de evaluación previamente diseñada, así se busca lograr que los productos informáticos desarrollados en la industria del software cumplan con criterios de alta calidad orientados a estándares de codificación.

2. Estándares de programación

Los estándares de programación son una parte integral de un software, según MacKellar intenta introducir este concepto a través de la aplicación y la educación mediante el documento de normas de programación. Se proporciona en medio de conferencias integradas con ejemplos adecuadamente, los estándares definidos en el documento y las asignaciones mejoradas para el proceso de desarrollo llamado “código de revisión por pares”, el cual se centra en la capacidad que tiene el estudiante para innovar dentro de un conjunto de restricciones definidas externamente que tienen como objetivo simular un ambiente de negocio[2].

Al concepto de estándares de programación se considera importante esta línea de seguimiento ya que se opta por especificar que estándares de desarrollo son adecuados para el módulo que se quiere realizar, no obstante, se hace una pregunta importante.

¿Porque es importante tener Buenas practicas al escribir código?

La realimentación hacia una evaluación continua y automática a través de las prácticas de programación proporciona información inmediata al desarrollador de los errores cometidos en codificación, permitiendo que puedan ser corregidas y reevaluadas repetidas veces, estableciendo un marco de evaluación donde la persona pueda aumentar su eficiencia en el momento de estar sentado escribiendo código. Según J. C. Rodríguez, estos son los siguientes errores por el cual son susceptibles para ser identificados inmediatamente

- Incumplimiento de interfaz
- Errores de estilo
- Diseño de codificación
- Errores de funcionalidad
- Errores graves de ejecución
- Insuficiencias en las pruebas[3]

Un ejemplo de la utilización de estándares de programación entorno a las empresas es asegurar que los programadores profesionales requieran normas de codificación más seguras y confiables, convirtiéndose en una situación que es muy frecuentemente uno de los problemas más importantes y que a menudo son descuidados en las empresas, así mismo cuando se tiene malos hábitos de escribir mal el código, por ejemplo, no comentar y sin embargo no conocer ningún tipo de estándares de programación, no se podrá entender de manera correcta lo que un desarrollador desea plasmar en su creación de software, tal es el caso que puede haber requisitos de codificación dependientes del lenguaje que requieran atención especial para evitar errores en la funcionalidad un programa.

Para empezar con las normas y reglas de estandarización del lenguaje de PHP primero se realiza una pregunta:

¿Qué es un PSR?

PSR (PHP Standards Recommendations) Es un acrónimo de la recomendación de normas de PHP lo cual contienen dentro de su formato PSR-1, PSR-2, PSR-3 y PSR-4 mediante el cual son las recomendaciones de PHP-FIG. Sus nombres comienzan con PSR- y terminan con un número. Cada recomendación de PHP-FIG resuelve un problema específico que con frecuencia se encuentra en la mayoría de los frameworks de PHP.

A. PSR-1: Estilo básico de código

Esta sección de la norma considera los elementos de codificación estándar proporciona pautas sencillas que son fáciles de implementar con un mínimo esfuerzo, debe cumplir estos requisitos para ser compatible con PSR-1:

- Etiquetas PHP: Lo cual sugiere redondear el código escrito en PHP con las etiquetas `<? Php?>` o `<? =?>`
- Codificación: Todos los archivos de PHP deben codificarse con el conjunto de caracteres UTF-8 que, por tal caso, un editor de texto más estructurado y completo lo hace automáticamente.
- Objetivo: Un solo archivo PHP puede definir símbolos (una clase, rasgo, función, constante, etc.) o realizar una acción que tiene efectos secundarios (por ejemplo, crear salida o manipular datos). Un archivo PHP no debería hacer ambas cosas. Esta es una tarea sencilla y requiere sólo un poco de previsión y planificación de su parte.
- Nombres de clases: el formato para el nombre de clases es *CamelCase*, refiriéndose al estilo de escritura que se aplica a frases o palabras compuestas, esto significa que el primer carácter del nombre del método está en minúsculas y la primera letra de cada palabra subsiguiente en el nombre del método es mayúscula.
- Nombres de las constantes: las constantes de PHP deben utilizar todos los caracteres en mayúsculas y utilizar subrayados para separar palabras.
- Nombres de métodos: de igual manera que los nombres de clase, los métodos deben tener el formato *CamelCase*.

B. PSR-2: Estilo de código estricto

Después de implementar PSR-1, el siguiente paso es implementar PSR-2. El estándar PSR-2 define el estilo de código PHP con pautas más estrictas. Un estilo de código estricto común permite a los desarrolladores escribir código que es fácilmente y rápidamente entendido por otros contribuyentes. A diferencia del PSR-1, la recomendación del PSR-2 contiene directrices más estrictas.

- Cada línea de código no debe exceder los 80 caracteres.
- Cada línea no debe tener espacio en blanco.
- Palabras claves en minúscula.
- Los espacios de nombres deben ir seguidos de una línea en blanco.
- **Clases:** La recomendación PSR-2 indica que el soporte de apertura de una definición de clase debe residir en una nueva línea inmediatamente después del nombre de definición de clase
- **Métodos:** La colocación de corchetes de definición de método es la misma que la colocación de corchetes de definición de clase.
- **Vistas:** Debe declarar una visibilidad para cada propiedad de clase y método. Una visibilidad es pública, protegida o privada; La visibilidad determina cómo una propiedad o método es accesible dentro y fuera de su clase.
- **Estructuras de control:** todas las palabras clave de la estructura de control deben ir seguidas de un solo carácter de espacio. Una palabra clave de estructura de control es **if, elseif, else, switch, case, while, do while, for, foreach, try o catch**. Si la palabra clave de estructura de control requiere un conjunto de paréntesis, asegúrese de que el primer paréntesis no esté seguido por un carácter de espacio y asegúrese de que el último paréntesis no esté precedido por un carácter de espacio.

C. PSR-3: interfaz Logger

PSR-3 es una recomendación y prescribe métodos que pueden ser implementados por los componentes del registrador de PHP, describiendo una interfaz común para registrar bibliotecas y escribir registros de una manera simple y universal.

Un componente logger de PHP compatible con la recomendación del PSR-3 debe incluir:

- Una clase PHP que implemente la interfaz denominada.
- Un componente Monolog PHP que implementa completamente la interfaz PSR-3, y se amplía fácilmente con formateadores de mensajes personalizados y controladores.

- Un manejador de mensajes de Monolog que permite enviar mensajes de registro a archivos de texto, syslog, correo electrónico, HipChat, Slack, servidores en red, APIs remotas, bases de datos y prácticamente cualquier otro lugar que pueda imaginar.
- En el evento muy improbable Monolog no proporciona un controlador para el destino de salida deseado, lo cual es super fácil escribir e integrar su propio controlador de mensajes Monolog.

D. PSR-4: Autoloaders

¿Qué pasa si necesita incluir un millar de scripts PHP? o ¿Qué pasa si necesita incluir un centenar de scripts PHP? Las funciones require () e include () no escalan bien y esta es la razón por la cual los autoloaders de PHP son importantes. Un autoloader es una estrategia para encontrar una clase, interfaz o rasgo de PHP y cargarlo en el intérprete de PHP bajo demanda en tiempo de ejecución, sin incluir explícitamente los archivos describiendo una estrategia de autocargador estandarizada.

Los componentes y frameworks de PHP que soportan el estándar del autocargador PSR-4 pueden ser localizados y cargados en el intérprete de PHP con un solo autocargador [4].

3. Métrica de evaluación

Observando lo anteriormente expuesto sobre estándares de codificación y los beneficios que brinda el utilizar estándares al momento de realizar desarrollos de software, es importante analizar cada una de las directrices que propone el estándar explicado como lo es el PSR. Este estándar propone una serie de normas que aplicarlas al momento de codificar software permite que este sea más limpio y pueda ser fácilmente entendido por otro programador; Analizando las directrices que expone el estándar, se busca identificar cuáles de estas son de mayor utilidad y tienen mayor peso para lograr el objetivo de obtener un código de calidad y que pueda ser fácilmente mantenido.

Partiendo de los supuestos anteriores después de identificar que directrices son de mayor importancia, se busca elaborar la métrica de evaluación teniendo como categorías o elementos de evaluación cada una de las normas del estándar especificado y asignándole un porcentaje de calificación de acuerdo a la importancia o peso considerado; posteriormente se definirá los niveles de calificación para cada una de las categorías, estos deben ser cuantificables y medibles, con esto se busca generar una calificación la cual pueda estar basada en normas y estándares usados en la industria de software actual.

Con el diseño de esta métrica se busca poder evaluar al programador para verificar si está aplicando el estándar de la forma correcta, además el poder ser calificado permite al programador poder identificar en qué aspectos está presentando falencias al momento de codificar, con esto se busca lograr que el software entregado sea mucho mejor elaborado.

Para la construcción de la métrica de evaluación las categorías que se van a evaluar son algunas de las directrices propuestas por el estándar, las cuales son:

- Declaración de variables.
- Declaración de métodos.
- Declaración de clases.
- Indentación de las estructuras de control.
- Uso de comentarios en el código fuente.
- Declaración de constantes.
- Uso de espacios de nombre.

A cada una de estas directrices se le asigna una prioridad de acuerdo a su grado de relevancia dentro del estándar propuesto, con esto se busca que las directrices que son usadas con mayor frecuencia y que pueden ser de mayor importancia para lograr el correcto uso del estándar tengan un mayor porcentaje al momento de obtener la calificación final.

Tabla50: Prioridad de cada directriz.

Directrices	Prioridad		
	Alta	Media	Baja
Variables	X		
Métodos	X		
Clases	X		
Indentación		X	
Comentarios		X	
Constantes			X
Espacios de nombre			X

Observando la tabla anterior se pueden identificar tres niveles de prioridades alta, media y baja. Cada una de estas tendrá un valor numérico que estará en el rango de 1 a 5 siendo las directrices de prioridad alta las que posean un mayor valor.

3.1 Ecuaciones

La forma de obtener el índice de cohesión final se basará en obtener la calificación individual de cada una de las normas, para luego consolidarlas en una única ecuación y obtener la nota correspondiente, para hacer este cálculo se utiliza una ecuación para cada una de las directrices anteriormente mencionadas, las cuales se presentan a continuación:

$$Nv = \frac{\sum_{N=0}^{NTA} \frac{NVC}{NVT}}{NTA} \quad (8)$$

NTA: Número total de archivos de código de fuente donde se hayan declarado variables.

NVC: Numero de variables correctamente declaradas.

NVT: Numero de variables totales declaradas por archivo de código fuente.

Nv: Calificación para la directriz declaración de variables.

$$Nm = \frac{\sum_{N=0}^{NTA} NMC}{NTA} \quad (9)$$

NTA: Número total de archivos de código de fuente donde se hayan declarado métodos.
NMC: Numero de métodos correctamente declarados.
NMT: Numero de métodos totales declarados por archivo de código fuente.
Nm: Calificación para la directriz declaración de métodos.

$$Nc = \frac{\sum_{N=0}^{NTA} NCC}{NTA} \quad (10)$$

NTA: Número total de archivos de código de fuente donde se hayan declarado clases.
NCC: Numero de clases correctamente declarados.
NCT: Numero de clases totales declaradas por archivo de código fuente.
Nv: Calificación para la directriz declaración de clases.

$$Ni = \frac{\sum_{N=0}^{NTA} NIC}{NTA} \quad (11)$$

NTA: Número total de archivos de código de fuente.
NIC: Numero de estructuras de control correctamente indentadas.
NIT: Numero de estructuras de control utilizadas por archivo de código fuente.
Ni: Calificación para la directriz de indentación de estructuras de control.

$$Ncm = \frac{\sum_{N=0}^{NTA} CMC}{NTA} \quad (12)$$

NTA: Número total de archivos de código de fuente.
CMC: Numero de estructuras de código comentadas.
CMT: Numero de estructuras utilizadas por archivo de código fuente.
Ncm: Calificación para la directriz uso de comentarios en el código fuente.

$$Ncs = \frac{\sum_{N=0}^{NTA} CSC}{NTA} \quad (13)$$

NTA: Número total de archivos de código de fuente donde se hayan declarado constantes.
CSC: Numero de constantes correctamente declarados.
CST: Numero de constantes totales declaradas por archivo de código fuente.

Ncs: Calificación para la directriz declaración de constantes.

$$Ne = \frac{\sum_{N=0}^{NTA} \frac{NSC}{NST}}{NTA} \quad (14)$$

NTA: Número total de archivos de código de fuente donde se hayan declarado espacios de nombre.

NSC: Numero de espacios de nombre correctamente declarados.

NST: Numero de espacios de nombre declarados por archivo de código fuente.

Ne: Calificación para la directriz de uso de espacios de nombre.

El cálculo del índice final se realizara aplicando una única ecuación en la cual se consolidaran cada una de las calificaciones obtenidas para cada directriz y estas tendrán un peso dentro de la ecuación de acuerdo a su a nivel de prioridad.

$$Ic = \frac{Vp(Nv) + Vp(Nm) + Vp(Nc) + Vp(Ni) + Vp(Ncm) + Vp(Ncs) + Vp(Ne)}{Vp(Npa) + Vp(Npm) + Vp(Npb)} \quad (15)$$

Vp: Valor de la prioridad de cada directriz.

Npa: Número de directrices de prioridad alta.

Npm: Número de directrices de prioridad media.

Npb: Número de directrices de prioridad baja.

Ic: Índice de cohesión.

La ecuación (8) se utiliza para poder calcular el índice de cohesión, para esto se necesita el valor obtenido para cada una de las directrices anteriormente explicadas, por ello cada uno de estos valores se multiplica por su respectiva prioridad y se sumaran, para luego ser divididos por el total de las directrices utilizadas en la codificación. El índice de cohesión busca medir a que nivel se está utilizando el estándar al momento de codificar, este valor puede ser representado como un porcentaje y así poder evaluar al programador sobre la aplicación de esta norma de calidad.

4. Importancia de los estándares de programación en un software de calidad

Los documentos de estándares de programación, definen el proceso de codificación como prácticas de matrices hacia una clasificación lo cual se debe adherir los estudiantes al desarrollar código. Existe una norma ISO para los lenguajes de programación en C que permite la implementación por parte de los compiladores y así mismo no todos los compiladores cumplen los estándares de programación, al enseñar a los desarrolladores a diseñar código a un cierto estándar, se promoverán los conceptos de confiabilidad de código, facilidad de mantenimiento y portabilidad, ya que el estudiante tendrá la capacidad de estar preparado para un ambiente de negocios. "Un estudio demostró empíricamente que la violación de las normas de los estándares, es impacto negativo en un software de fiabilidad"[2].

Durante el paso de los años los estándares de codificación se han vuelto populares como un medio para asegurar la calidad de software durante el proceso de desarrollo de un software, se aseguran un estilo común de programación, lo que aumenta la capacidad de mantenimiento y evitan el uso de construcciones potencialmente generadoras de problemas, aumentando así la confiabilidad. Las reglas en tales estándares se basan principalmente en la opinión experta y ganada por los años con una cierta lengua en varios contextos.

Los autores Zhenmin Li y Lin Tan en su investigación “Un estudio empírico de las características de los errores en el software de código abierto moderno” argumentan que los primeros controles automatizados han contribuido a la fuerte disminución de los errores de memoria presentes en el software. Sin embargo, la disponibilidad de normas y herramientas adecuadas, existen varias cuestiones hacen obstáculo, Por ejemplo, el 30% de las líneas de uno de los proyectos utilizados en este estudio contenían tal violación. Las infracciones pueden ser subproductos del análisis estático, que no siempre puede determinar si el código viola o no una determinada comprobación[5].

Tomando otro ejemplo como referencia, encontramos que una sola regla es responsable del 83% de todas las violaciones en uno de los proyectos analizados, lo cual es poco probable que solo señale los verdaderos problemas.

“Cualquier modificación del software tiene una probabilidad no nula de introducir un nuevo fallo, y si esta probabilidad excede la reducción obtenida al fijar la infracción, el resultado neto es una mayor probabilidad de fallas en el software”[6].

El término plagio en el software es difícil de distinguir, Sin embargo, a través del uso de esta norma específicamente en las secciones que tratan con comentarios e identificadores, se hace evidente y más fácil distinguir entre el diseño de problemas de grupo que comparte (que es el aprendizaje) y la copia de código. Además, el uso de los exámenes y revisiones de código de compañeros resaltarán a los estudiantes que no intentan aprender.

El análisis de requisitos para el diseño se captura en el campo de comentario como lo hace la noción de gestión de configuración y plan de prueba de verificación proporcionando al estilo de diseño de código un estándar para la legibilidad, comprensión y facilidad de mantenimiento. Para evaluar el conocimiento de los aprendices sobre el análisis de los requisitos a través del diseño, el uso y la corrección de los comentarios son fundamentales. Esto se ajusta a las mejores prácticas, Las revisiones de código se utilizan como proceso para reforzar la capacidad de mantenimiento y la conformidad[2].

Finalmente los aspectos más representativos de un estándar de codificación hace limitar al lenguaje a un conjunto seguro para evitar el uso de instrucciones inseguras, reduciendo eficazmente los defectos del código y mejorando la calidad de software, una de las normas de codificación ampliamente aceptadas para el software es MISRA-C. MISRA-C es un conjunto de directrices para el uso de programación en este caso en C, primeramente, publicado en 1998 y realizando así una nueva versión en 2004, contiene 141 reglas de estándares de programación conveniente para el desarrollo crítico. Aborda directamente un diseño inadecuado para mejorar la capacidad del sistema, ofreciendo un servicio con fiable y logrando de esta manera ser aceptado por industrias y aplicaciones como lo son: militar, automóvil, aeroespacial, medico etc.

Para la comprobación de reglas relacionadas con las variables MISRA-C desensamblada el código fuente siendo implícito lo que hace que logre representar las variables de maneras operadas en código de desensamble, como resultado puede colocarse en un registro general o en una pila y referenciada por su

dirección, siendo totalmente diferente al código fuente. Una variable de código fuente hace desaparecer el código de desensamble debido a las optimizaciones del compilador, todo esto significa que algunas reglas de estándar de codificación con las variables no pueden ser verificadas[7].

5. Conclusiones

A través del escrito hecho, se puede llegar a dos interrogantes importantes, que concluyen la escritura de este artículo.

¿Qué reglas usar para el estándar?

Un estándar puede ser simplemente ampliamente adoptado, pero todavía contienen reglas que, a simple vista no parecen las más adecuadas para un proyecto, además la simple existencia de diferentes estándares de codificación, se evalúa explícitamente las reglas individuales que pueden ayudar a valorar el estándar correcto.

¿Cómo priorizar una lista de violaciones?

Aunque un estándar puede ser cuidadosamente elaborado y personalizado para un proyecto, los desarrolladores se enfrentan a demasiadas violaciones fijas, dado el tiempo limitado. Para manejar este problema de la manera más eficiente, se hace énfasis en una lista de violaciones, para definir un umbral para determinar cuáles deben ser abordados, y cuáles pueden omitirse.[8].

¿Por qué es importante implementar un estándar de codificación como lo es el PSR?

Hoy en día se hace relativamente difícil en seguir correctamente los estándares de programación especialmente en extensos proyectos de software; el cumplimiento de una norma de codificación se trata en este caso una prueba de aprobación o rechazo, sin embargo, es posible que se convierta en un enfoque diferente donde se mide el nivel de cumplimiento, esto permitirá correlacionar con las mediciones de los productos de software. Contar el número de violaciones de tales reglas en un producto de Software parece estar bien fundamentado e intuitivamente corresponde a una medida de su calidad interna.[9]

Por tal motivo en los trabajos de desarrollo en los que se ha participado, se identifica que, al realizar una colaboración con una mayor cantidad de miembros en el equipo de trabajo, cada uno de estos tiene un estilo de codificación totalmente diferente al de los demás, esto logra dificultar y entorpecer la comprensión y análisis del código, por lo tanto una recomendación que logra mitigar este tipo de inconvenientes es implementar normas de programación actuales, como lo es el PSR siendo un estándar actual utilizado por PHP y frameworks como laravel, para lograr un estilo uniforme en cada uno de los scripts que componen el proyecto, dando la impresión que han sido escritos por un solo desarrollador, con esto se busca que el mantenimiento de un producto de software sea totalmente entendible para el grupo de desarrollo que trabajara sobre este en futuro.

Referencias

[1] P. Roa, C. Morales, and P. Gutierrez, "Norma ISO / IEC 25000," vol. 3, no. 2, 2015.

- [2] D. V. MacKellar, "Injection of Business Coding Standards Practices to Embedded Software Courses Donald," in *2016 IEEE Frontiers in Education Conference (FIE)*, 2016, pp. 1–8.
- [3] J. C. Rodríguez, D. Pino, M. D. Roca, Z. H. Figueroa, and D. González Domínguez, "Hacia la Evaluación Continua Automática de Prácticas de Programación."
- [4] J. Lockhart, *Modern PHP New Features and Good Practices*. 2015.
- [5] Z. Li, L. Tan, X. Wang, S. Lu, Y. Zhou, and C. Zhai, "Have things changed now?," *Proc. 1st Work. Archit. Syst. Support Improv. Softw. dependability - ASID '06*, pp. 25–33, 2006.
- [6] C. Boogerd and L. Moonen, "Using software history to guide deployment of coding standards," *Trader Reliab. high-volume Consum. Prod.*, pp. 39–52, 2009.
- [7] Z. Dai, X. Mao, D. Wang, D. Liu, and J. Zhang, "Checking compliance to coding standards for x86 executables," *Proc. - Symp. Work. Ubiquitous, Auton. Trust. Comput. Conjunction with UIC 2010 ATC 2010 Conf. UIC-ATC 2010*, no. November 2010, pp. 449–455, 2010.
- [8] C. Boogerd and L. Moonen, "Evaluating the relation between coding standard violations and faults within and across software versions," *Proc. 2009 6th IEEE Int. Work. Conf. Min. Softw. Repos. MSR 2009*, pp. 41–50, 2009.
- [9] W. Basalaj Co- and F. Van Den Beuken, "CORRELATION BETWEEN CODING STANDARDS COMPLIANCE AND SOFTWARE QUALITY."

Anexo D. Artículo transferencia colcaci 2018

SOFTWARE APLICADO A LA MEDICIÓN DE ESTÁNDARES DE CODIFICACIÓN UTILIZANDO UN MODELO MATEMÁTICO PARA CALCULAR EL ÍNDICE DE COHESIÓN EN LOS PRODUCTOS DE DESARROLLO WEB

Cesar Yesid Barahona Rodríguez
Héctor Hernán Castellanos Rodríguez
John Fredy Osorio Franco

*Universidad de Cundinamarca
Facatativá, Colombia*

cbarahona@ucundinamarca.edu.co

hccastellanos@ucundinamarca.edu.co

jfosorio@ucundinamarca.edu.co

Abstract— En busca de lograr automatizar procesos en el desarrollo de software se plantea crear un software que permita realizar el proceso de evaluación de los estándares de codificación en los ficheros de código fuente. Para esto se investiga sobre los estándares que existen en la actualidad, sus beneficios y características, además de esto se diseña un métrica de evaluación basada en un modelo matemático, para que esta sea utilizada por el software para poder calcular el índice de cohesión entre el software desarrollado y el código evaluado para poder determinar si el código cumple con las normas establecidas por el estándar seleccionado en esta investigación.

Keywords—Estándares de codificación, calidad de software, mantenibilidad, integridad de código.

I. INTRODUCCIÓN

En la industria del software uno de los aspectos que representa un mayor costo en el proceso de construcción de software, es el de realizar mantenimiento al código fuente, esto puede deberse a varios aspectos, como pueden ser el no utilizar una arquitectura adecuada, el no seguir un proceso de desarrollo adecuado, el no aplicar estándares de

codificación, entre otras razones. En este artículo se busca enfatizar en este último aspecto, el no utilizar estándares de codificación, en muchas fábricas de software, no se exige a los desarrolladores que apliquen un estándar en específico, si no se deja la libertad total al desarrollador de codificar de la manera que considere correcta, es por esto que cuando es necesario realizar cambios sobre el mismo se dificulta este proceso debido a la variación que puede existir en cada uno de los ficheros que componen el proyecto, a razón de esto en ocasiones es necesario reescribir el fichero al no entender el mismo lo que duplica el tiempo y costo del desarrollo.

En busca de mitigar este aspecto se realiza un proceso investigativo y de desarrollo, con la cual se quiere resaltar la importancia de usar un estándar de codificación y a través de esta investigación encontrar un estándar que sea utilizado en la actualidad, esto con el fin de diseñar un modelo matemático que permita calcular un índice de cohesión del software, para proseguir con el desarrollo de un software que permita calcular este índice en los ficheros de código fuente escritos por los desarrolladores y así poder emitir conclusiones sobre el mismo para poder proceder a la aprobación o rechazo del mismo.

II. IMPORTANCIA DE LOS ESTÁNDARES DE PROGRAMACIÓN EN UN SOFTWARE DE CALIDAD

Los documentos de estándares de programación definen el proceso de codificación como prácticas de matrices hacia una clasificación lo cual se debe adherir los estudiantes al desarrollar código. Existe una norma ISO para los lenguajes de programación en C que permite la implementación por parte de los compiladores y así mismo no todos los compiladores cumplen los estándares de programación, al enseñar a los desarrolladores a diseñar código a un cierto estándar, se promoverán los conceptos de confiabilidad de código, facilidad de mantenimiento y portabilidad, ya que el estudiante tendrá la capacidad de estar preparado para un ambiente de negocios.

“Un estudio demostró empíricamente que la violación de las normas de los estándares, es impacto negativo en un software de fiabilidad”[1].

Durante el paso de los años los estándares de codificación se han vuelto populares como un medio para asegurar la calidad de software durante el proceso de desarrollo de un software, se aseguran un estilo común de programación, lo que aumenta la capacidad de mantenimiento y evitan el uso de construcciones potencialmente generadoras de problemas, aumentando así la confiabilidad. Las reglas en tales estándares se basan principalmente en la opinión experta y ganada por los años con una cierta lengua en varios contextos.

Los autores Zhenmin Li y Lin Tan en su investigación “Un estudio empírico de las características de los errores en el software de código abierto moderno” argumentan que los primeros controles automatizados han contribuido a la fuerte disminución de los errores de memoria presentes en el software. Sin embargo, la disponibilidad de normas y herramientas adecuadas, existen varias cuestiones hacen obstáculo, Por ejemplo, el 30% de las líneas de uno de los proyectos utilizados en este estudio contenían tal violación. Las infracciones pueden ser subproductos del análisis estático, que no siempre puede determinar si el código viola o no una determinada comprobación.[2]

“Cualquier modificación del software tiene una probabilidad no nula de introducir un nuevo fallo, y si esta probabilidad excede la reducción obtenida al fijar la infracción, el resultado neto es una mayor probabilidad de fallas en el software”[3].

El análisis de requisitos para el diseño se captura en el campo de comentario como lo hace la noción de gestión de configuración y plan de prueba de verificación proporcionando al estilo de diseño de código un estándar para la legibilidad, comprensión y facilidad de mantenimiento. Para evaluar el conocimiento de los aprendices sobre el análisis de los requisitos a través del diseño, el uso y la corrección de los comentarios son fundamentales. Esto se ajusta a las mejores prácticas, Las revisiones de código se utilizan como proceso para reforzar la capacidad de mantenimiento y la conformidad.[1]

Final mente los aspectos más representativos de un estándar de codificación hace limitar al lenguaje a un conjunto seguro para evitar el uso de instrucciones inseguras, reduciendo eficazmente los defectos del código y mejorando la calidad de software, una de las normas de codificación ampliamente aceptadas para el software es MISRA-C.

MISRA-C es un conjunto de directrices para el uso de programación en este caso en C, primeramente, publicado en 1998 y realizando así una nueva versión en 2004, contiene 141 reglas de estándares de programación conveniente para el desarrollo crítico. Aborda directamente un diseño inadecuado para mejorar la capacidad del sistema, ofreciendo un servicio confiable y logrando de esta manera ser aceptado por industrias y aplicaciones como lo son: militar, automóvil, aeroespacial, medico etc.

Para la comprobación de reglas relacionadas con las variables MISRA-C desensamblada el código fuente siendo implícito lo que hace que logre representar las variables de maneras operadas en código de desensamble, como resultado puede colocarse en un registro general o en una pila y referenciada por su dirección, siendo total mente diferente al código fuente. Una variable de código fuente hace desaparecer el código de desensamble debido a las optimizaciones del compilador, todo esto significa que algunas reglas de estándar de codificación con las variables no pueden ser verificadas.[4]

III. MÓDULO PARA LA EVALUACIÓN DE ESTÁNDARES DE CODIFICACIÓN BAJO LA METODOLOGÍA DE CALIDAD DE SOFTWARE

Algunas razones por la cual se espera que los desarrolladores de la universidad de Cundinamarca adopten y sigan los estándares de codificación no es solo la familiarización con el proceso de desarrollo, si no que se promueva la legibilidad, transparencia, organización y portabilidad del código fuente, con el propósito que el código pueda mantenerse, reutilizarse y reducir costos de desarrollo de software. Beneficiando a los grupos de desarrolladores o a una organización más grande para reducir tiempo requerido por las personas para comprender o revisar el trabajo de sus compañeros.

Es por eso que el módulo que se presenta a continuación Como apoyo a la actividad investigativa desarrollada en el programa de ingeniería de sistemas, extensión Facatativá es un anexo al software de calidad llamado Calisoft el cual se fundamenta en una plataforma de evaluación para los productos de software que se desarrollan en la universidad que se basan en la realización de software mediante tres (3) sistemas de calificaciones. El primer sistema de calificación es en donde se presenta al evaluador la documentación y modelación del proyecto, el segundo sistema es la parte en donde se evalúa la gestión de pruebas tanto funcionales como prueba de carga y estrés, la tercera herramienta cuenta con la parte administrativa en donde se hacen las configuraciones de acuerdo con los estándares de calidad.

El módulo que se desarrollo es un complemento al sistema de evaluación de gestión de pruebas, lo cual mejorara el sistema de calificación de la plataforma de Calisoft, este complemento dará un gran soporte ante los estándares de programación que se rigen actualmente en el mundo, se busca que con la implementación de este módulo los evaluadores puedan garantizar que se esté respetando el estándar de codificación previamente establecido, mientras que el módulo de proyecto de estandarización determine una calificación hacia el desarrollador, de esta manera se puede constatar en que posibles errores o en qué aspectos el desarrollador está fallando a la hora de codificar, recordando la importancia que tiene este ítem para garantizar un software de calidad.

Así mismo para la garantía de la realización de este módulo se basó en 3 puntos importantes

- Investigación previa con el objetivo de realizar una documentación y seleccionar un estándar de codificación adecuado
- Construcción de una Métrica de evaluación
- Desarrollo del módulo para la evaluación de estándares de codificación

IV. CRITERIOS DE CODIFICACION

Antes de empezar primero se realizó una investigación predefiniendo unos requisitos principales para saber la instancia y la magnitud del proyecto los cuáles son

- Generar los parámetros de estándares
- Evaluar declaración de variables, clases, funciones, contantes, indentación, declaración de comentarios y espacios de nombre.
- Creación de la métrica de evaluación con las anteriores directrices para crear un sistema de calificaciones de manera cuantitativa.
- Creación de la interfaz gráfica.
- Visualización del código fuente del desarrollador.
- Realización de informes de calificación del módulo de evaluación de estándares de codificación.

Los desarrollos de los productos de software deben estar soportados por normas de calidad que hayan sido emitidas por organizaciones internacionales como la ISO teniendo en cuenta que la evaluación generada cumpla con las normas internacionales como la ISO/IEEE 25010, buscando establecer las pautas mínimas que debe cumplir un software de calidad. La norma ISO/IEEE 25010 establece un modelo de calidad de software, un sistema de calificación de los productos, determinando las características de calidad que se tienen en cuenta a la hora de

evaluar las propiedades de un producto de software determinado.

V. MÉTRICA DE EVALUACIÓN

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad.

- Adecuación función
- Eficiencia de desempeño
- Compatibilidad
- Usabilidad
- Fiabilidad
- Seguridad
- Mantenibilidad
- Portabilidad

El cual los productos de software están ligados por la parte de la mantenibilidad el cual representa su modificación efectiva y eficiente debido a las necesidades evolutivas o perspectivas.[5]

Por último, se escogió un estándar de codificación base que por el cual los estudiantes de la universidad de Cundinamarca deben realizar su respectiva lectura, análisis y seguimiento para poder realizar una evaluación correspondiente de sus productos de software correspondientes a la codificación son las recomendaciones del estándar de PHP (PHP Standards Recommendations) Es un acrónimo de la recomendación de normas de PHP lo cual contienen dentro de su formato PSR-1, PSR-2, PSR-3 y PSR-4 mediante el cual son las recomendaciones de PHP-FIG. Sus nombres comienzan con PSR- y terminan con un número. Cada recomendación de PHP-FIG resuelve un problema específico que con frecuencia se encuentra en la mayoría de los frameworks de PHP[6].

- PSR-1: Estilo básico de código
- PSR-2: Estilo de código estricto
- PSR-3: interfaz Logger
- PSR-4: Autoloaders.

Observando lo anteriormente expuesto sobre estándares de codificación y los beneficios que brinda el utilizar estándares al momento de realizar desarrollos de software, es importante analizar cada una de las directrices que propone el estándar explicado como lo es el PSR. Este estándar propone una serie de normas que aplicarlas al momento de codificar nuestro software permite que este sea más limpio y pueda ser fácilmente entendido por otro programador; Analizando las directrices que expone el estándar, se busca identificar cuáles de estas son de mayor utilidad y tienen mayor peso para lograr el objetivo de obtener un código de calidad y que pueda ser fácilmente mantenido.

Partiendo de los supuestos anteriores después de identificar que directrices son de mayor importancia, se busca elaborar la rúbrica de evaluación teniendo como categorías o elementos de evaluación cada una de las normas del estándar especificado y asignándole un porcentaje de calificación de acuerdo a la importancia o peso considerado, posteriormente se definirá los niveles de calificación para cada una de las categorías, estos deben ser cuantificables y medibles, con esto se busca generar una calificación la cual pueda estar basada en normas y estándares usados en la industria de software actual.

Con el diseño de esta métrica se busca poder evaluar al programador para verificar si está aplicando el estándar de la forma correcta, además el poder ser calificado permite al programador poder identificar en qué aspectos está presentando falencias al momento de codificar, con esto se busca lograr que el software entregado sea mucho mejor elaborado.

Para la construcción de la métrica de evaluación las categorías que se van a evaluar son algunas de las directrices propuestas por el estándar, las cuales son:

- Declaración de variables.
- Declaración de métodos.
- Declaración de clases.
- Indentación de las estructuras de control.
- Uso de comentarios en el código fuente.
- Declaración de constantes.
- Uso de espacios de nombre.

A cada una de estas directrices se le asigna una prioridad de acuerdo con su grado de relevancia dentro del estándar propuesto, con esto se busca que las directrices que son usadas con mayor frecuencia y que

pueden ser de mayor importancia para lograr el correcto uso del estándar tengan un mayor porcentaje al momento de obtener la calificación final.

Directrices	Prioridad		
	Alta	Media	Baja
VARIABLES	X		
MÉTODOS	X		
CLASES	X		
INDENTACIÓN		X	
COMENTARIOS		X	
CONSTANTES			X
ESPACIOS DE NOMBRE			X

Tabla 51. Prioridad de cada directriz.

Observando la tabla anterior se pueden identificar tres niveles de prioridades alta, media y baja. Cada una de estas tendrá un valor numérico que estará en el rango de 1 a 5 siendo las directrices de prioridad alta las que posean un mayor valor.

La forma de realizar el cálculo de la calificación final se basará en obtener la nota individual de cada una de las normas, para luego consolidarlas en una única ecuación y obtener la nota correspondiente, para hacer este cálculo se utiliza una ecuación para cada una de las directrices anteriormente mencionadas, las cuales se presentan a continuación:

$$Nv = \frac{\sum_{N=1}^{NTA} NVC}{NTA} \quad (16)$$

NTA: Número total de archivos de código de fuente donde se hayan declarado variables.

NVC: Numero de variables correctamente declaradas.

NVT: Numero de variables totales declaradas por archivo de código fuente.

Nv: Calificación para la directriz declaración de variables.

$$Nm = \frac{\sum_{N=1}^{NTA} NMC}{NTA} \quad (17)$$

NTA: Número total de archivos de código de fuente donde se hayan declarado métodos.

NMC: Numero de métodos correctamente declarados.

NMT: Numero de métodos totales declarados por archivo de código fuente.

Nm: Calificación para la directriz declaración de métodos.

$$Nc = \frac{\sum_{N=1}^{NTA} NCC}{NTA} \quad (18)$$

NTA: Número total de archivos de código de fuente donde se hayan declarado clases.

NCC: Numero de clases correctamente declarados.

NCT: Numero de clases totales declaradas por archivo de código fuente.

Nv: Calificación para la directriz declaración de clases.

$$Ni = \frac{\sum_{N=1}^{NTA} NIC}{NTA} \quad (19)$$

NTA: Número total de archivos de código de fuente.

NIC: Numero de estructuras de control correctamente indentadas.

NIT: Numero de estructuras de control utilizadas por archivo de código fuente.

Ni: Calificación para la directriz de indentación de estructuras de control.

$$N_{cm} = \frac{\sum_{N=1}^{NTA} \frac{CMC}{CMT}}{NTA} \quad (20)$$

NTA: Número total de archivos de código de fuente.

CMC: Numero de estructuras de código comentadas.

CMT: Numero de estructuras utilizadas por archivo de código fuente.

Ncm: Calificación para la directriz uso de comentarios en el código fuente.

$$N_{cs} = \frac{\sum_{N=1}^{NTA} \frac{CSC}{CST}}{NTA} \quad (21)$$

NTA: Número total de archivos de código de fuente donde se hayan declarado constantes.

CSC: Numero de constantes correctamente declarados.

CST: Numero de constantes totales declaradas por archivo de código fuente.

Ncs: Calificación para la directriz declaración de constantes.

$$N_e = \frac{\sum_{N=1}^{NTA} \frac{NSC}{NST}}{NTA} \quad (22)$$

NTA: Número total de archivos de código de fuente donde se hayan declarado espacios de nombre.

NSC: Numero de espacios de nombre correctamente declarados.

NST: Numero de espacios de nombre declarados por archivo de código fuente.

Ne: Calificación para la directriz de uso de espacios de nombre.

El cálculo de la nota final se realizara aplicando una única ecuación en la cual se consolidarán cada una de las notas obtenidas para cada directriz y estas tendrán

un peso dentro de la ecuación de acuerdo a su nivel de prioridad.

$$\frac{Vp(Nv) + Vp(Nm) + Vp(Nc) + Vp(Ni) + Vp(Ncm) + Vp(Ncs) + Vp(Ne)}{Vp(Npa) + Vp(Npm) + Vp(Npb)} \quad (23)$$

Vp: Valor de la prioridad de cada directriz.

Npa: Número de directrices de prioridad alta.

Npm: Número de directrices de prioridad media.

Npb: Número de directrices de prioridad baja.

La ecuación (8) se utiliza para poder calcular el índice de cohesión, para esto se necesita el valor obtenido para cada una de las directrices anteriormente explicadas, por ello cada uno de estos valores se multiplica por su respectiva prioridad y se sumaran, para luego ser divididos por el total de las directrices utilizadas en la codificación. El índice de cohesión busca medir a que nivel se está utilizando el estándar al momento de codificar, este valor puede ser representado como un porcentaje y así poder evaluar al programador sobre la aplicación de esta norma de calidad.

VII. DESARROLLO Y FUNCIONAMIENTO DEL MODULO

En la etapa final de este trabajo de investigación se realizara el desarrollo de una herramienta que tome como base el estándar que se seleccionó anteriormente y evalúe de acuerdo a la métrica expuesta en el presente artículo, los archivos de código fuente que componen el proyecto que se esté desarrollando; permitiendo calcular el índice de cohesión, para poder emitir conclusiones de si el desarrollo cumple con las medidas de calidad deseadas para el producto software.

El funcionamiento de esta herramienta se enfoca en el análisis de código fuente para determinar si el estándar fue aplicado en un nivel, que sea aceptable para las normas de calidad deseadas para el producto. Es por esto que esta herramienta tendrá una arquitectura lineal la cual incluirá varios componentes, estos a su vez tendrán como salida objetos que serán la entrada

para el siguiente componente, esto con el fin de procesar el código y poder calcular el índice de cohesión para el proyecto.

Los componentes en los cuales está dividida la herramienta buscan filtrar y clasificar los elementos del código para procesarlos individualmente y evaluarlos de acuerdo con el estándar.

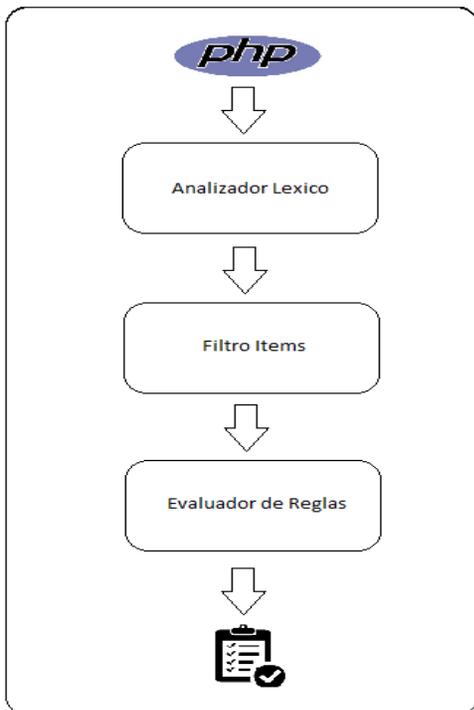


Fig. 54. Arquitectura del módulo

Como se puede observar en la Figura 1. La herramienta está constituida por tres componentes, cada uno de estos recibe un elemento de entrada y a su vez entregará otro que servirá de entrada para el componente siguiente. Cada uno de estos componentes tiene un funcionamiento y objetivo diferente los cuales se presentan a continuación:

A. Analizador Léxico.

Este componente recibe como entrada el código fuente que está escrito en lenguaje PHP. El proceso que realiza es reconocer las palabras y elementos que hacen parte del léxico del lenguaje, a su vez se encarga de clasificarlos en

grupos, según la función que cumplen dentro del lenguaje. Cada vez que el analizador lee un elemento nuevo, crea un token el cual es un objeto que contiene información acerca de ese elemento como el grupo en que se clasifico, valor y demás información que será necesaria para los procesos siguientes, una vez que el token ha sido creado se anexa a lista de tokens.

Una vez que el analizador termina de realizar la lectura del fichero, procede a entregar la lista de tokens creada, al siguiente componente el cual se encargara de continuar el proceso.

B. Filtro Ítems

Este componente recibe como entrada la lista de tokens entregada por el componente anterior. Este se encarga de filtrarlos de acuerdo al ítem o directriz de evaluación especificados en la métrica de evaluación. A medida que filtra cada uno de los tokens los anexa a una estructura ordenada y elimina los demás elementos que no son necesarios para la evaluación. Una vez termina de recorrer la lista, este componente se encarga de entregar al siguiente una estructura reducida y ordenada para que este realice el proceso final de evaluación.

C. Evaluador de Reglas.

El proceso de este componente inicia recibiendo del anterior, una estructura ordenada de acuerdo con los ítems de evaluación establecidos. Esta toma cada uno de los elementos de la estructura, verifica a que directriz de evaluación pertenece y realiza el proceso de evaluación según sean las normas establecidas para esa directriz, si este componente aprueba las normas de evaluación se procede a computarlo según se estableció en la métrica de evaluación. Una vez este termina la validar cada una de los elementos, genera un resultado de evaluación y lo entrega como salida del módulo.

Este proceso se repite para cada uno de los ficheros que se van a evaluar, una vez que se han evaluado todos, se realizara el cálculo del índice de cohesión el cual es el objetivo final de todo este proceso.

VIII. CONCLUSIONES

Se pudo identificar qué beneficios trae el utilizar estándares de codificación como el incremento en la fiabilidad del software, la disminución en el tiempo de ejecución, entre otros. A su vez se pudo seleccionar un estándar de codificación como lo es el PSR para continuar el proceso de investigación.

La construcción de un modelo matemático que permita calcular el índice de cohesión se pudo realizar implementando un esquema basado en directrices jerarquizadas en prioridades, basándose en su frecuencia de uso e importancia dentro del estándar.

El desarrollo del módulo se pudo realizar implementado una arquitectura lineal y utilizando herramientas de software libre. Un siguiente paso en este proceso será realizar pruebas al mismo sometiéndolo a diferentes escenarios de prueba para determinar el nivel de certeza que este pueda brindar.

REFERENCIAS

- [1] D. V. MacKellar, "Injection of Business Coding Standards Practices to Embedded Software Courses Donald," in *2016 IEEE Frontiers in Education Conference (FIE)*, 2016, pp. 1–8.
- [2] Z. Li, L. Tan, X. Wang, S. Lu, Y. Zhou, and C. Zhai, "Have things changed now?," *Proc. 1st Work. Archit. Syst. Support Improv. Softw. dependability - ASID '06*, pp. 25–33, 2006.
- [3] C. Boogerd and L. Moonen, "Using software history to guide deployment of coding standards," *Trander Reliab. high-volume Consum. Prod.*, pp. 39–52, 2009.
- [4] Z. Dai, X. Mao, D. Wang, D. Liu, and J. Zhang, "Checking compliance to coding standards for x86 executables," *Proc. - Symp. Work. Ubiquitous, Auton. Trust. Comput. Conjunction with UIC 2010 ATC 2010 Conf. UIC-ATC 2010*, no. November 2010, pp. 449–455, 2010.
- [5] P. Roa, C. Morales, and P. Gutierrez, "Norma ISO / IEC 25000," *Univ. Dist. Fr. Jose Caldas*, vol. 3, no. 2, 2015.
- [6] J. Lockhart, *Modern PHP New Features and Good Practices*. 2015.
- [7] C. Boogerd and L. Moonen, "Evaluating the relation between coding standard violations and faults within and across software versions," *Proc. 2009 6th IEEE Int. Work. Conf. Min. Softw. Repos. MSR 2009*, pp. 41–50, 2009.

Anexo F. Reportes pruebas de calidad



UDEC
UNIVERSIDAD DE
CUNDINAMARCA

CALISOFT

Calle 14 con Avenida 15
Universidad de Cundinamarca - Ext. Facativá
(+57 1) 892 0706 | 892 0707
unicundi@ucundinamarca.edu.co

estándares de codificación - Evaluación de codificación

Nombre 9975_ItemsEvaluadosController.php			
Items	Total	Acertadas	Nota
Variables	3	2	0.67
Clases	1	1	1
Funciones	2	2	1
Constantes	0	0	0
Identacion	3	3	1
Comentarios	4	4	1
Espacios De Nombre	1	1	1

Observación

Nombre 7057_ItemsCodificacionController.php			
Items	Total	Acertadas	Nota
Variables	2	2	1
Clases	1	1	1
Funciones	2	2	1
Constantes	0	0	0
Identacion	3	2	0.67
Comentarios	4	4	1
Espacios De Nombre	1	1	1

Nombre 7057_ItemsCodificacionController.php			
Items	Total	Acertadas	Nota

Observación

Nombre 7724_NotaScriptController.php			
Items	Total	Acertadas	Nota
Variables	3	2	0.67
Clases	1	1	1
Funciones	3	2	0.67
Constantes	0	0	0
Identacion	4	4	1
Comentarios	5	5	1
Espacios De Nombre	1	1	1

Observación

Nombre 2931_ScriptController.php			
Items	Total	Acertadas	Nota
Variables	7	7	1
Clases	1	1	1
Funciones	4	4	1
Constantes	0	0	0
Identacion	5	4	0.8
Comentarios	6	6	1
Espacios De Nombre	1	1	1

Observación

Nombre 2754_ItemsCodificacion.php			
Items	Total	Acertadas	Nota

Nombre		2754_ItemsCodificacion.php		
Items	Total	Acertadas	Nota	
Variables	3	3	1	
Clases	3	3	1	
Funciones	2	2	1	
Constantes	0	0	0	
Identacion	2	2	1	
Comentarios	3	3	1	
Espacios De Nombre	1	1	1	

Observación

Nombre		2699_ItemsEvaluados.php		
Items	Total	Acertadas	Nota	
Variables	3	3	1	
Clases	3	2	0.67	
Funciones	2	2	1	
Constantes	0	0	0	
Identacion	2	2	1	
Comentarios	3	3	1	
Espacios De Nombre	1	1	1	

Observación

Nombre		1084_Script.php		
Items	Total	Acertadas	Nota	
Variables	4	4	1	
Clases	4	4	1	
Funciones	3	3	1	

Nombre		1084_Script.php	
Items	Total	Acertadas	Nota
Constantes	0	0	0
Identacion	3	3	1
Comentarios	4	3	0.75
Espacios De Nombre	1	1	1
Observación			
Promedio General (Codificación)		86.9%	



UDEC
UNIVERSIDAD DE
CUNDINAMARCA

CALISOFT

Calle 14 con Avenida 15
Universidad de Cundinamarca - Ext. Facativá
(+57 1) 892 0706 | 892 0707
unicundi@ucundinamarca.edu.co

estándares de codificación - Evaluación de modelado

Diagrama de clases.pdf		Diagrama de clases (100%)	
Evaluador	Paisa		
Promedio	100%		
Componente	Validación	Observación	
Notación de clase	Correcto		
Interfaces	Correcto		
Tablas	Correcto		
Asociaciones	Correcto		
Generalizaciones	Correcto		
Agregaciones	Correcto		
Clase asociación	Correcto		
Dependencias	Correcto		
Trazado	Correcto		
Relaciones	Correcto		
Anidamientos	Correcto		

Casos de uso.pdf		Diagrama de casos de uso (100%)	
Evaluador	Paisa		
Promedio	100%		
Componente	Validación	Observación	

Evaluador	Paiza	
Promedio	100%	
Componente	Validación	Observación
Actores	Correcto	
Casos de uso	Correcto	
Inclusión de casos de uso	Correcto	
Casos de uso extendidos	Correcto	
Puntos de extensión	Correcto	
Límite del sistema	Correcto	

Diagramas de secuencia.pdf Diagrama de :

Evaluador	Paiza	
Promedio	82%	
Componente	Validación	Observación
Línea de vida	Correcto	
Mensajes	Correcto	
Ocurrencia de ejecución	Correcto	
Mensajes self	Correcto	
Mensajes perdidos y encontrados	Incorrecto	
Inicio y final de línea de vida	Correcto	
Restricciones de tiempo y duración	Correcto	
Fragmentos combinados	Correcto	
Puerto	Incorrecto	
Descomposición en parte	Correcto	
Continuaciones / Invariantes de Estado	Correcto	

Diagramas de actividades.pdf Diagrama de

Evaluador	Paiza	
Promedio	93%	
Componente	Validación	Observación
Actividad	Correcto	
Acción	Correcto	
Restricciones de acción	Correcto	
Flujo de control	Correcto	
Nodo inicial	Correcto	
Nodo final de actividad	Correcto	
Nodo final de flujo	<i>Incorrecto</i>	
Flujo de objetos	Correcto	
Nodos de decisión y combinación	Correcto	
Nodos de bifurcación y unión	Correcto	
Región de expansión	Correcto	
Gestores de excepción	Correcto	
Región de actividad interrumpible	Correcto	
Partición	Correcto	

Promedio General Modelacion	94%
-----------------------------	-----



UDEC
UNIVERSIDAD DE
CUNDINAMARCA

CALISOFT

Calle 14 con Avenida 15
Universidad de Cundinamarca - Ext. Facatativá
(+57 1) 892 0706 | 892 0707
unicundi@ucundinamarca.edu.co

estándares de codificación - Evaluación de Base de Datos

Componente	Total	Acertadas	Calificacion
Base de Datos	1	1	5
Esquemas	0	0	0
Tablas	31	29	4.68
Vistas	0	0	0
Llaves Primarias	29	29	5
Llaves Foraneas	30	30	5
Campo Descripcion	0	0	0
Campo ValorMoneda	0	0	0
Campo Observaciones	0	0	0
Obsevacion	calificado		
Calificacion Total	5		



UDEC
UNIVERSIDAD DE
CUNDINAMARCA

CALISOFT

Calle 14 con Avenida 15
Universidad de Cundinamarca - Ext. Facatativá
(+57 1) 892 0706 | 892 0707
unicundi@ucundinamarca.edu.co

estándares de codificación - Resultados

Modelacion	94
Plataforma	0
Codificacion	87
Base de Datos	98
Total	93