

	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 5

26.

FECHA	jueves, 31 de mayo de 2018
--------------	----------------------------

Señores
UNIVERSIDAD DE CUNDINAMARCA
 BIBLIOTECA
 Ciudad

UNIDAD REGIONAL	Extensión Facatativá
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
Marquéz Rangel	Stevenson	1073172384
Suárez Campos	Johan Camilo	1030638504

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

APELLIDOS COMPLETOS	NOMBRES COMPLETOS
Barahona Rodríguez	Cesar Yesid

TÍTULO DEL DOCUMENTO
REINGENIERÍA DE LA PLATAFORMA CALISOFT PARA LA IMPLEMENTACIÓN DE MÉTODO DE TESTING NO HEURÍSTICO PARA LA EVALUACIÓN DE CALIDAD DE PRODUCTOS SOFTWARE

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



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 5

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

AÑO DE EDICIÓN DEL DOCUMENTO	NÚMERO DE PÁGINAS
31/05/2018	205p

DESCRIPTORES O PALABRAS CLAVES EN ESPAÑOL E INGLÉS (Usar 6 descriptores o palabras claves)	
ESPAÑOL	INGLÉS
1. Automatización	Automation
2. Ingeniería de la usabilidad	Usability engineering
3. Validación	Validation
4. Calidad	Quality
5. Testeo	Testing
6. Métodos no heurísticos	Non-heuristic methods

RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS
(Máximo 250 palabras – 1530 caracteres, aplica para resumen en español):

Este libro expone como se utilizaron métodos de Testing no heurísticos para la calificación de productos software en la universidad de Cundinamarca; los métodos no heurísticos evaluarán la usabilidad del software, con el objetivo de que el producto final a evaluar sea más íntegro y comprensible para los diferentes usuarios. Estos métodos se integran en CALISOFT (Calidad de software), una plataforma de la universidad de Cundinamarca (la cual se le hizo un proceso de reingeniería) destinada a evaluar la calidad de productos software. Se implementaron los métodos no heurísticos a los diferentes módulos de CALISOFT, con el objetivo de realizar una evaluación más precisa, rápida y con un grado mayor de eficacia. Los métodos no heurísticos están programados, lo que hace que los productos se evalúen bajo un mismo estándar. Se usó una metodología de investigación mixta para evaluar los resultados obtenidos y así determinar la importancia y el cambio que otorgó la implementación de estos métodos.

This book exposes how non-heuristic Testing methods are used for software qualification at the University of Cundinamarca; The non-heuristic methods evaluate the usability of the software, with the aim of making the final product larger and more comprehensible for different users. These methods are integrated in CALISOFT (Software Quality), a platform of the University of Cundinamarca, destined to the evaluation of the software of products. The non-heuristic methods were implemented in the different CALISOFT modules, with the aim of making a more precise, faster and more efficient evaluation. The nonheuristic methods are programmed, which means that the products are evaluated under the same standard. A mixed research methodology was used to evaluate the results obtained and thus determine the importance and the change that the implementation of these methods granted.



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 5

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) 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.



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 5

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:

- 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.

	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 5

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.



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



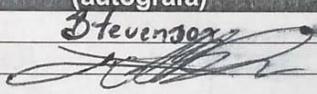
Nota:

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. Reingeniería de la plataforma CALISOFT para la implementación de método de testing no heurístico para la evaluación de calidad de productos software.pdf	PDF
2.Formato de repositorio institucional.pdf	PDF
3.	
4.	

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

APELLIDOS Y NOMBRES COMPLETOS	FIRMA (autógrafa)
Marqu3ez Rangel Stevenson	
Su3arez Campos Johan Camilo	

12.1.50

**REINGENIERÍA DE LA PLATAFORMA CALISOFT PARA LA
IMPLEMENTACIÓN DE MÉTODO DE TESTING NO HEURÍSTICO PARA LA
EVALUACIÓN DE CALIDAD DE PRODUCTOS SOFTWARE**

STEVENSON MÁRQUEZ RANGEL

JOHAN CAMILO SUÁREZ CAMPOS

UNIVERSIDAD DE CUNDINAMARCA

Facultad de Ingeniería

Programa de Ingeniería de Sistemas

Facatativá, Abril de 2018

**IMPLEMENTACIÓN DE MÉTODO DE TESTING NO HEURÍSTICO PARA LA
EVALUACIÓN DE CALIDAD DE PRODUCTOS SOFTWARE**

STEVENSON MÁRQUEZ RANGEL

JOHAN CAMILO SUÁREZ CAMPOS

**Director: César Yesid Barahona Rodríguez, Ingeniero en
Telecomunicaciones, Especialista en Gestión de Proyectos**

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

UNIVERSIDAD DE CUNDINAMARCA

Facultad de Ingeniería

Programa de Ingeniería de Sistemas

Facatativá, Abril de 2017

Gracias a todos nuestros docentes y compañeros que nos apoyaron con cada proceso de nuestra carrera profesional, "Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad." Albert Einstein.

COMPROMISO DE AUTORES

Yo, Stevenson Márquez Rangel con célula de identidad No. 1073172384 y con código 461214135. estudiante del programa 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:

Yo, Johan Camilo Suárez Campos con célula de identidad No. 1030638504 y con cód. 461214173 estudiante del programa 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

Este libro expone como se utilizaron métodos de Testing no heurísticos para la calificación de productos software en la universidad de Cundinamarca; los métodos no heurísticos evaluarán la usabilidad del software, con el objetivo de que el producto final a evaluar sea más íntegro y comprensible para los diferentes usuarios. Estos métodos se integran en CALISOFT (Calidad de software), una plataforma de la universidad de Cundinamarca (la cual se le hizo un proceso de reingeniería) destinada a evaluar la calidad de productos software.

Se implementaron los métodos no heurísticos a los diferentes módulos de CALISOFT, con el objetivo de realizar una evaluación más precisa, rápida y con un grado mayor de eficacia. Los métodos no heurísticos están programados, lo que hace que los productos se evalúen bajo un mismo estándar. Se usó una metodología de investigación mixta para evaluar los resultados obtenidos y así determinar la importancia y el cambio que otorgó la implementación de estos métodos.

Palabras clave: Automatización, ingeniería de la usabilidad, validación, calidad, Testing, integración de módulos, métodos no heurísticos, reingeniería.

This book exposes how non-heuristic Testing methods are used for software qualification at the University of Cundinamarca; The non-heuristic methods evaluate the usability of the software, with the aim of making the final product larger and more comprehensible for different users. These methods are integrated in CALISOFT (Software Quality), a platform of the University of Cundinamarca, destined to the evaluation of the software of products.

The non-heuristic methods were implemented in the different CALISOFT modules, with the aim of making a more precise, faster and more efficient evaluation. The non-heuristic methods are programmed, which means that the products are evaluated under the same standard. A mixed research methodology was used to evaluate the results obtained and thus determine the importance and the change that the implementation of these methods granted.

Key words: Automation, usability engineering, validation, quality, Testing, module integration, non-heuristic methods, reengineering.

ÍNDICE GENERAL

RESUMEN.....	14
INTRODUCCIÓN.....	19
I. INFORME DE INVESTIGACIÓN.....	20
1.1. ESTADO DEL ARTE.....	20
1.2. LÍNEA DE INVESTIGACIÓN.....	21
1.3. GRUPO DE INVESTIGACIÓN.....	21
1.4. PLANTEAMIENTO DEL PROBLEMA.....	22
1.5. OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS.....	22
1.6. IMPACTO DEL PROYECTO.....	23
1.7. METODOLOGÍA.....	23
1.8. MARCOS DE REFERENCIA.....	24
1.8.1. MARCO TEÓRICO.....	24
1.8.2. MARCO LEGAL.....	25
1.8.1. MARCO HISTÓRICO.....	26
II. DOCUMENTACIÓN DEL SOFTWARE.....	27
2.1. PLAN DE PROYECTO.....	27
2.2. DETERMINACIÓN DE REQUERIMIENTOS.....	28
2.3. ESPECIFICACIÓN DEL DISEÑO.....	30
2.3.1. MODELO DE IDENTIDAD RELACIÓN (MER).....	31
2.3.2. DIAGRAMAS DE CASOS DE USO.....	32
2.3.3. DIAGRAMAS DE SECUENCIAS.....	34
2.3.4. DIAGRAMA DE ACTIVIDADES.....	39
2.3.5. DIAGRAMA DE CLASES.....	43
2.3.6. DIAGRAMA DE COLABORACIÓN.....	44
2.3.7. DIAGRAMA DE DESPLIEGUE.....	45
2.4. DISEÑO DE CASOS DE PRUEBA.....	46
2.5. ESTIMACIÓN DE RECURSOS.....	51
2.6. RESULTADOS.....	53
2.7. CONCLUSIONES Y RECOMENDACIONES.....	61
2.8. BIBLIOGRAFÍA.....	63
2.9. ANEXOS.....	64
2.9.1. MANUAL DE INSTALACIÓN.....	64
2.9.2. MANUAL DE USUARIO.....	103
2.9.3. ARTÍCULO 1.....	200
2.9.4. ARTÍCULO 2.....	205
2.9.5. CRONOGRAMA.....	213

LISTA DE TABLAS

Tabla 1 Requerimientos del sistema.....	29
Tabla 2 Disyunción exclusiva XOR	30
Tabla 3 Descripción Modelo Entidad Relación	32
Tabla 4 Descripción Roles, casos de uso	33
Tabla 5 Ejecutar testeo, casos de uso	33
Tabla 6 Evalúa campos de entrada, casos de uso	34
Tabla 7 Ingresa valores correctos, casos de uso.....	34
Tabla 8 Ingresa valores correctos, casos de uso.....	34
Tabla 9 Descripción secuencia ejecutar testeo.....	35
Tabla 10 Descripción secuencia evalúa campos de entrada	36
Tabla 11 Descripción secuencia ingresa valores correctos	37
Tabla 12 Descripción ingresa valores incorrectos.....	37
Tabla 13 descripción secuencia general.....	39
Tabla 14 Descripción actividad ejecutar testeo.....	40
Tabla 15 Descripción evalúa campos de entrada	41
Tabla 16 Descripción actividad ingresa valores correctos	41
Tabla 17 Descripción actividad ingresa valores incorrectos	41
Tabla 18 Descripción actividades centrales.....	43
Tabla 19 Descripción diagrama de clases	44
Tabla 20 Descripción diagrama de colaboración	45
Tabla 21 Descripción diagrama de despliegue	46
Tabla 22 Financiación.....	51
Tabla 23 Resumen por rubros	51
Tabla 24 Detalle de personal	52
Tabla 25 Descripción de equipos.....	52
Tabla 26 Descripción de materiales e insumos.....	52
Tabla 27 Descripción de servicios tecnológicos.....	53
Tabla 28 Descripción de viajes	53
Tabla 29 Descripción de otros	53

LISTA DE GRÁFICOS

Gráfico 1 Modelo Procesos de Monitoreo y Control de Pruebas	26
Gráfico 2 Diagrama de entidad relación (MER)	31
Gráfico 3 Diagrama de casos de uso	33
Gráfico 4 Diagrama de secuencia, ejecutar testeo	35
Gráfico 5 Diagrama de secuencia, evalúa campos de entrada.....	36
Gráfico 6 Diagrama de secuencia, ingresar valores correctos.....	36
Gráfico 7 Diagrama de secuencia, ingresa valores incorrectos	37
Gráfico 8 Diagrama de secuencia, central	38
Gráfico 9 Diagrama de actividades, ejecutar testeo.....	40
Gráfico 10 Diagrama de actividades, evaluar campos de entrada.....	40
Gráfico 11 Diagrama de actividades, ingresar valores correctos	41
Gráfico 12 Diagrama de actividades, ingresar valores incorrectos	41
Gráfico 13 Diagrama de actividades, general	42
Gráfico 14 Diagrama de clases.....	44
Gráfico 15 Diagrama de colaboración.....	44
Gráfico 16 Diagrama de despliegue.....	45

LISTA DE IMÁGENES

Imagen 1 Validación campo nombre caso prueba, evaluador	46
Imagen 2 Validación en campos de caso prueba, evaluador.....	47
Imagen 3 Validación de fecha límite, evaluador	48
Imagen 4 Validación de JSON, desarrollador	48
Imagen 5 Validación de fecha límite	49
Imagen 6 Funcionamiento de módulo de Testing	50
Imagen 7 Validación en tiempo real.....	50
Imagen 8 Prueba guardada	51
Imagen 9 Página principal del antiguo Calisoft	54
Imagen 10 Dashboard de evaluador del antiguo Calisoft	54
Imagen 11 Página principal del nuevo Calisoft	55
Imagen 12 Dashboard de evaluador del nuevo Calisoft	55
Imagen 13 Vista casos de prueba, antiguo Calisoft	56
Imagen 14 Vista casos de prueba, nuevo Calisoft	56
Imagen 15 Crear caso prueba, antiguo Calisoft.....	57
Imagen 16 Crear caso prueba, nuevo Calisoft.....	57
Imagen 17 Uso de la extensión, nuevo Calisoft	58
Imagen 18 Subir formulario, nuevo Calisoft	59
Imagen 19 Subir formulario, antiguo Calisoft	59
Imagen 20 Módulo de Testing, antiguo Calisoft	60
Imagen 21 Módulo de Testing, nuevo Calisoft	60
Imagen 22 Procedimiento de Testing automatizado, nuevo Calisoft	61

INTRODUCCIÓN

Actualmente en la Universidad de Cundinamarca existe un software llamado CALISOFT, basado en la norma ISO/IEC/IEEE 29119, que sirve como apoyo en el proceso de evaluación de calidad de módulos de proyectos basados en aplicativos web. El software CALISOFT se sometió a un proceso de reingeniería por parte de un grupo de trabajo de la Universidad de Cundinamarca, donde los autores de este libro fueron partícipes, el proceso consistió en volver a reestructurar el software bajo las mismas normas ISO/IEC/IEEE 29119, con el objetivo mejorar los procesos internos, procesos externos, rediseñar arquitectura y bases de datos, crear un ambiente más amigable con el usuario final y dejar una estructura que favorezca la integración futura de otros módulos.

La estructura inicial de CALISOFT está compuesta por tres partes fundamentales, la primera parte consiste en la modelación y documentación de los proyectos a evaluar, la segunda parte se centra en la gestión de pruebas (Testing), y una tercera parte que es un módulo administrativo. El objetivo de este proceso de investigación es mejorar el módulo de gestión de pruebas, lo que se denomina Testing, además de integrar dos módulos externos destinados a valorar la nomenclatura y modelación de la base de datos, y evaluar el estándar de codificación de los productos de software presentados.

El módulo de Testing de CALISOFT es realizado por un desarrollador de la Universidad de Cundinamarca, este evalúa el aplicativo bajo su propio criterio, el cual se rige por el número de inserción de datos que el desarrollador efectuó en los diferentes campos de texto que presente el aplicativo, y la calidad evaluada irá conforme a la cualidad que tenga el desarrollador evaluador de someter la página a diferentes tipos de entradas de datos, como por ejemplo; ingresar varios correos escritos de diferentes formas o ingresar datos erróneos o no lógicos. Esta tarea se puede tornar extensa y agotadora para el evaluador, ya que en un aplicativo puede tener una gran cantidad y variedad de campos de entradas distintos y a eso se le debe sumar que cada campo se evalúa con al menos diez datos distintos para comprobar su eficacia. En este proceso de reingeniería se implementará la ingeniería de la usabilidad en el módulo de Testing, simulando una evaluación no heurística por medio de métodos programados, esta simulación de técnicas no heurísticas hace un testeo más riguroso y eficaz, ya que se aumentan el número de intentos a la hora de ingresar datos en un campo de texto, al igual que aumenta la variedad de posibles datos de entrada. El tiempo de procesamiento se reduce bastante comparado con el anterior CALISOFT, otorgando que la calidad se evalúe con mayor precisión. Los dos módulos externos se implementaron en CALISOFT, se modificó la interfaz de una forma más amigable para el usuario final. Su estructura interna es simple, para que en un futuro si se desea incorporar un nuevo módulo, sea mucho más sencillo y práctico. Su estructura externa es rápida ya que el usuario final no necesita estar navegando en múltiples vistas para realizar una sola operación.

I. INFORME DE INVESTIGACIÓN

1.1. ESTADO DEL ARTE

El ser humano siempre ha querido optimizar su trabajo, lo que llevo a la creación de las maquinas, y con el pasar del tiempo el hombre ha ido mejorándolas para que su usabilidad sea más óptima y sencilla, la usabilidad se puede definir como “la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.”(M. Mascheroni, Greiner, & Petris, 2012).

“El término Ingeniería de Usabilidad se introduce por primera vez en Digital Equipment Corporation, para referirse al conjunto de conceptos y técnicas que permiten planificar, realizar y verificar los objetivos de la usabilidad de un sistema”. (Centro Nacional de Información de Ciencias Médicas & Moráquez Bergues, 2013).

La usabilidad se pronunció en el siglo XX, en donde se hizo notar los ordenadores y el internet, el crecimiento fue tanto que los diseñadores y desarrolladores de aplicaciones que usan dichos medios buscaban maneras de mejorar la experiencia del usuario a través de la usabilidad, para optimizar el diseño y conseguir un producto más competitivo en el mercado. Con ello la usabilidad se ha ido utilizando en aplicaciones informáticas o en otros sistemas que tengan que ver con una interacción directa con el usuario. (Bolaños-Pizarro & Vidal-Infer, A; Navarro-Molina, Carolina; Valderrama-Zuriañ, Juan Carlos; Aleixandre-Benavent, 2007).

La usabilidad tiene sus formas de evaluarse, una de ellas es el test con usuarios (Testing) es la técnica más utilizada en los análisis de usabilidad, la cual consiste en dar una serie de instrucciones o tareas a un grupo de usuarios destinados a probar páginas, con el fin de que registren todos los problemas que puedan encontrar. (Bolaños-Pizarro & Vidal-Infer, A; Navarro-Molina, Carolina; Valderrama-Zuriañ, Juan Carlos; Aleixandre-Benavent, 2007)

El Testing tiene dos métodos principales para su evaluación, método heurístico y método no heurístico (Empírico). De forma heurística cuando “es llevado a cabo por expertos en usabilidad o en Interacción Hombre-Computadora (HCI), basándose en su experiencia de diseño o en guías de diseño de usabilidad para señalar críticas sobre el sistema, Mientras que el método no heurístico lo conforman usuarios de conocimiento comunes.” (M. A. Mascheroni, Greiner, Dapozo, & Estayno, 2014)

Es crucial que para el momento en que se decida seleccionar un método de Testing, se deba tener en cuenta primero el costo de su implementación, las restricciones que se puedan presentar en el tiempo y la competitividad del método a utilizar. (Sánchez & Ovidio, 2013)

Con el rápido avance del internet han surgido múltiples empresas enfocadas hacia las pruebas de calidad de software implementando la usabilidad por medio del método evaluativo Testing. Entre esas empresas están las siguientes:

Según la asociación (Expo, 2014) Globe Testing, es una compañía de nivel internacional de servicios especializada en las pruebas de software y la monitorización de sistemas. Con sede central en Berlín, Globe opera en Alemania, Suiza y España, dando servicio a grandes y medianas empresas en varios sectores verticales.

Sogeti España, perteneciente al Grupo Capgemini, es una compañía especializada en servicios y soluciones de tecnología, focalizada en Testing y Calidad de

Software, Soluciones Microsoft y High Tech Consulting. El Grupo Sogeti es líder mundial en Testing y Calidad de Software, cuenta con más de 12.000 profesionales dedicados específicamente a esta área en 15 países de Europa, EE.UU. y la India. (Expo, 2014)

Sogeti lleva 15 años ofreciendo servicios de Testing y Calidad de Software en el mercado español. Sogeti es el creador de las metodologías estándar del mercado: TMap® (Test Management Approach) y TPI® (Test Process Improvement). Además, es autor de numerosas publicaciones sobre Calidad de Software, entre ellas el informe anual "World Quality Report", y creador del evento "Testing & Tools Day".(Expo, 2014)

También ha habido numerosos aportes de parte de la comunidad "open source" en forma de marcos de trabajo que facilitan y normalizan el medio en el que el desarrollador realiza las pruebas necesarias en su producto de software, las diversas tecnologías que podemos encontrar actualmente son las siguientes:

Mocha es un framework de pruebas y un ejecutor de test que ejecuta código JavaScript en un entorno Node.js, siendo este el principal caso de uso para Mocha: testing a bajo nivel de proyectos Node. Para este propósito, el de ejecutar test a bajo nivel, Mocha es el mejor ejecutor de test posibles. Como añadido a esto, Mocha es capaz de ser controlado en su totalidad a través de la línea de comandos, además de poder ser fácilmente integrable en diferentes IDEs actuales. (Carmona, 2016)

Selenium es un entorno de pruebas de software para aplicaciones basadas en la web. En la actualidad, la gran mayoría de aplicaciones software que son desarrolladas, están pensadas para que estas sean ejecutadas en un navegador web. (Carmona, 2016).

Una de las características más importantes de este entorno es la capacidad que tiene para poder las pruebas en múltiples plataformas, desde navegadores de escritorio, hasta navegadores de plataformas móviles. (Carmona, 2016)

1.2. LÍNEA DE INVESTIGACIÓN

Software, sistemas emergentes y nuevas tecnologías.

1.3. GRUPO DE INVESTIGACIÓN

Como apoyo a la actividad investigativa desarrollada en el programa de ingeniería de sistemas, extensión Facatativá, se crea con aval de Colciencias, en el año 2010 el grupo de investigación de sistemas y tecnología de Facatativá (GISTFA). Sobre el año 2012 GISTFA abandona el concepto de grupo de estudio con el que nació para dar paso a la creación de propuestas serias hacia la solución de problemas.

Es así como GISTFA asume la responsabilidad de la investigación del programa con la tarea de convertir los resultados de sus semilleros en proyectos eficientes, óptimos y de impacto en la comunidad académica y en su entorno. Sobre esta base y bajo convenios hechos con las empresas de la región GISTFA ha implementado proyectos de desarrollo e innovación en la Alcaldía Municipal, Escuela de Comunicaciones, Policía Nacional y la empresa de aguas de Facatativá. Fue relevante para el grupo los trabajos desarrollados con el I.D.R.D del distrito Capital entre otras. Otros intereses de la investigación en el grupo GISTFA lo comprometen hacia una labor en la cual un ejercicio permanente actualice su razón de ser a partir de proyectos informáticos que generen conocimiento pertinente y consistente en el

tiempo. Ahora bien, el grupo GISTFA crece y así las investigaciones orientadas por el grupo han contribuido a la creación de ambiciosas metas como la generación de proyectos interdisciplinarios (en el caso de la alianza con el grupo de investigación AXIOMA y la transición hacia el emprendimiento y la innovación para fortalecer la transferencia de conocimiento y las conexiones con el sector empresarial.

Actualmente y desde su labor investigativa, el grupo GISTFA apoya al CIT (Centro de Innovación y Tecnologías) del programa de Ingeniería de Sistemas, Facatativá, con el objetivo de aumentar el impacto de los productos generados por el grupo de investigación GISTFA en la evaluación realizada por COLCIENCIAS y así mismo proponer nuevas metodologías de desarrollo como resultado de los productos generados a lo largo de la vida de los proyectos informáticos.

1.4. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, la calidad de los productos software de la universidad de Cundinamarca se evalúan con la plataforma “CALISOFT” cuyo sistema de evaluación se basa en el almacenamiento de documentos que posteriormente son revisados de forma manual por el evaluador, siendo este un docente de la facultad. Dicha labor resulta extensa ya que en los proyectos de mayor complejidad o de características específicas, requieren numerosas pruebas por cada módulo del aplicativo, ya que evaluar un solo proyecto requiere un tiempo y esfuerzo de trabajo humano; el proyecto actual busca que esta tarea sea simplificada con asistencia de la misma plataforma, que ejecutará métodos no heurísticos programados para que la tarea de comprobación de calidad sea efectuada en menor tiempo, y la rigurosidad de las pruebas dependan menos de la habilidad del que las realiza. Además, se integrarán dos módulos externos que pretenden ser hincapié en la evaluación de calidad.

¿Actualizar el software CALISOFT de la universidad de Cundinamarca Facatativá con métodos no heurísticos programados e integración módulos externos hará que las pruebas de calidad sean más eficaces?

1.5. OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS

OBJETIVO GENERAL

Actualizar el software CALISOFT con el fin de mejorar la evaluación de calidad que se empleen en los aplicativos web presentados.

OBJETIVOS ESPECÍFICOS

- Entender el método no heurístico para posteriormente implementarlo en el módulo de testeo de CALISOFT.
- Hacer que la labor de evaluación y calificación del software sea menos dependiente de la habilidad de la persona destinada a efectuar las pruebas.
- Modificar el modelo del software CALISOFT.
- Crear una interfaz gráfica de usuario que permita una mejor interacción con el usuario final, guiándolo en cada paso de manera intuitiva.
- Integrar dos módulos externos, que son; el módulo de análisis de código fuente y el módulo estándar de nomenclatura de base de datos. Con la finalidad de volver el software más integro, además de mejorar la certificación de calidad a un producto software que se evalúe por medio de la plataforma.

1.6. IMPACTO DEL PROYECTO

CALISOFT (calidad de software) es un software que evalúa aplicativos web desarrollados en la universidad de Cundinamarca, aprecia tanto modelación como funcionalidad.

La actualización de CALISOFT facilitará la acción de realizar las pruebas funcionales no heurísticas para módulos de proyectos basados en aplicativo web, como resultado hará que la labor de evaluación y calificación del software a tratar sea menos dependiente de la habilidad del evaluador destinado a realizar las pruebas, además realizará una evaluación más rigurosa y exacta.

CALISOFT definirá la calidad de los aplicativos web propuestos, esta labor será una tarea programada, lo que quiere decir que reducirá los tiempos de evaluación si se compara con el tiempo que le lleva a un humano en realizarlas, un humano se tarda normalmente 3 semanas en efectuar una prueba no heurística a un aplicativo web. De esta manera se espera reducir tiempos de resultados y aumentar el número de aplicativos a calificar por semana.

El proyecto también exigirá por parte del desarrollador un orden y un estándar de programación para certificar la eficacia del software a tratar, ya que se implementarán dos módulos externos para evaluar la calidad del aplicativo con un mayor índice. Esto a su vez será de ayuda para que el estudiante sepa los requisitos y características de un entregable de calidad.

1.7. METODOLOGÍA

Se implementará una metodología de investigación mixta para alcanzar los objetivos propuestos anteriormente. Esta metodología se caracteriza por integrar tanto el método cualitativo como el método cuantitativo usando las fortalezas de cada método con el fin de obtener un panorama más completo del fenómeno.

¿La integración de módulos externos al software CALISOFT lo hará más integral y eficaz al momento de evaluar la calidad de un producto software?

Para comprobar si la integración de módulos externos definirán con mayor exactitud la calidad de un producto software, se deberá evaluar conforme a un método mixto, en donde se investigará y analizará los factores principales que definen la calidad de un producto software, analizaremos a fondo cada característica y la compararemos con las nuevas características que nos aportarán los módulos a integrar, también se harán comparaciones con los módulos ya creados, se definirá que le hace falta a CALISOFT para ser integral y eficaz para definir la calidad de un producto software.

De este estudio se espera demostrar que los módulos a integrar harán de CALISOFT un software más integro a la hora de definir la calidad de un producto software.

¿Las técnicas no heurísticas programadas mejorarán la evaluación de calidad de aplicativos web?

Para comprobar si la implementación de técnicas no heurísticas programadas mejora la evaluación de calidad, se deberá evaluar con forme a un criterio cuantitativo, en donde se realizarán evaluaciones no heurísticas manuales y evaluaciones no heurísticas programadas en un periodo de una semana a un mismo aplicativo web. Los datos por obtener serán; cantidad de módulos evaluados,

cantidad de páginas evaluadas en cada módulo, cantidad de datos ingresados en cada módulo, cantidad de errores presentados en cada módulo. Este proceso se deberá repetir por dos semanas más y con un aplicativo web diferente por cada semana. Con los datos obtenidos se hará una comparación entre la labor del evaluador humano y el evaluador programado, donde se espera observar en los resultados una gran diferencia entre las cantidades de los datos de ambas evaluaciones. Con estos resultados obtenidos se podrá analizar y definir que método es más eficaz a para evaluar la calidad de un producto software.

1.8. MARCOS DE REFERENCIA

1.8.1.MARCO TEÓRICO

El ser humano siempre ha querido optimizar su trabajo, lo que llevo a la creación de las maquinas, y con el pasar del tiempo el hombre ha ido mejorándolas para que su usabilidad sea más óptima y sencilla, la usabilidad se puede definir como “la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.” (Mascheroni, Greiner, & Petris, 2012)

“El término Ingeniería de Usabilidad se introduce por primera vez en Digital Equipment Corporation, para referirse al conjunto de conceptos y técnicas que permiten planificar, realizar y verificar los objetivos de la usabilidad de un sistema”. (Centro Nacional de Información de Ciencias Médicas & Moráquez Bergues, 2013)

La usabilidad busca maneras de mejorar la experiencia del usuario para optimizar el diseño y conseguir un producto más competitivo en el mercado. Con ello la usabilidad se ha ido utilizando en aplicaciones informáticas o en otros sistemas que tengan que ver con una interacción directa con el usuario. (Bolaños-Pizarro & Vidal-Infer, A; Navarro-Molina, Carolina; Valderrrama-Zuriañ, Juan Carlos; Aleixandre-Benavent, 2007)

Las Organizaciones Internacionales de Estándares de Calidad como lo es ISO o IEEE, desenvuelven el concepto de usabilidad en diferentes normas, presentando la usabilidad como un atributo del software en cada una de ellas. En el estándar ISO 9241-11, es una norma que “hace referencia a la usabilidad como la medida en que un producto se puede usar por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso” (Moreno & Marciszack, 2013) La siguiente norma es, ISO/IEC 9126-1 que se encarga de definir a la usabilidad como “un parámetro de calidad del software y conforma una de las características de calidad relevantes del producto”. La norma ISO/IEC 9126-1 tienen como función principal la satisfacción de las necesidades del usuario por medio de los siguientes atributos: funcionalidad, fiabilidad, utilidad, eficiencia, mantenibilidad y portabilidad. Por último, la norma ISO/IEC 14598 “describe un proceso para evaluar la calidad de los productos de software, software, el cual es compatible con la norma anterior, y donde la usabilidad conforma la calidad del software”. (Moreno & Marciszack, 2013)

Actualmente en la Universidad de Cundinamarca existe un software llamado CALISOFT, basado en la norma ISO/IEC/IEEE 29119, que sirve como apoyo en el proceso de evaluación de calidad de módulos de proyectos basados en aplicativos web.

CALISOFT está compuesta por tres partes fundamentales, la primera parte consiste en la modelación y documentación de los proyectos a evaluar, la segunda parte se centra en la gestión de pruebas (Testing), y una tercera parte que es un módulo

administrativo. El objetivo en esta actualización es mejorar la parte de gestión de pruebas, lo que se denomina Testing, además de integrar dos módulos externos destinados a valorar la nomenclatura y modelación de la base de datos, y evaluar el estándar de codificación que presente el producto de software presentado.

La parte de Testing de CALISOFT es realizada por un desarrollador de la Universidad de Cundinamarca, este evalúa la página bajo su propio criterio, el cual se rige por el número de inserción de datos que el desarrollador efectuó en los diferentes campos de texto que presente el aplicativo, y la calidad evaluada irá conforme a la cualidad que tenga el programador de someter la página a diferentes tipos de entradas de datos, como por ejemplo; ingresar varios correos escritos de diferentes formas o ingresar datos erróneos o no lógicos. Esta tarea se puede tornar extensa y agotadora para el desarrollador, ya que en un aplicativo puede tener una gran variedad de campos de entradas distintos y a eso se le debe sumar que cada campo se evalúa con al menos 10 datos distintos para comprobar su eficacia. En esta actualización se implementará la ingeniería de la usabilidad en el módulo de Testing, simulando una evaluación no heurística por medio de métodos programados, esta simulación de técnicas no heurísticas hará un testeo más riguroso y eficaz, ya que se aumentarán los intentos a la hora de ingresar datos en un campo de texto, al igual que aumentará la variedad de posibles datos de entrada. Se espera que se reduzca el tiempo de respuesta y que la calidad se evalúe con mayor precisión. Los dos módulos externos se implementarán en CALISOFT, se modificará la interfaz y se adecuará un nuevo modelo con el fin de que acoplen de manera eficaz.

1.8.2.MARCO LEGAL

ISO / IEC / IEEE 29119 Software Testing es un conjunto de estándares internacionales usados para realizar pruebas de software en cualquier ciclo de vida de desarrollo de un software u organización, al usar e implementar estas normas se está adaptando los únicos estándares reconocidos internacionalmente para realizar pruebas en software, así otorgando una alta calidad en el manejo y uso de las pruebas de testeo. ("IEEE SA - 29119-1-2013 - Software and systems engineering —Software testing —Part 1: Concepts and definitions," 2013) Actualmente existen cinco estándares dentro de este grupo, de los cuales se harán uso los siguientes:

ISO / IEC / IEEE 29119-1 facilita el uso de las diferentes normas ISO / IEC / IEEE 29119, en el cual incluye los conceptos y el vocabulario en el que se construyen estas normas, además proporciona una serie de ejemplos aplicados en la práctica, en resumen éste estándar es más informativo, ayuda a tener en claro un punto de partida hacia los otros estándares. ("IEEE SA - 29119-1-2013 - Software and systems engineering —Software testing —Part 1: Concepts and definitions," 2013)

ISO / IEC / IEEE 29119-2 define los modelos de los procesos de las pruebas de software que se emplean en cualquier ciclo de desarrollo del software. El propósito de estos modelos es especificar con detalle los procesos de las pruebas que se utilicen al momento de evaluar un software. El proceso de las pruebas se basa en un modelo de tres capas, que son; especificaciones de la prueba, gestión de pruebas y pruebas dinámicas. El modelo de "procesos de monitoreo y control de pruebas" Figura 1. Es uno de los modelos de proceso de prueba que se define en ISO / IEC / IEEE 29119-2, el cual se implementará en el testeo no heurístico.

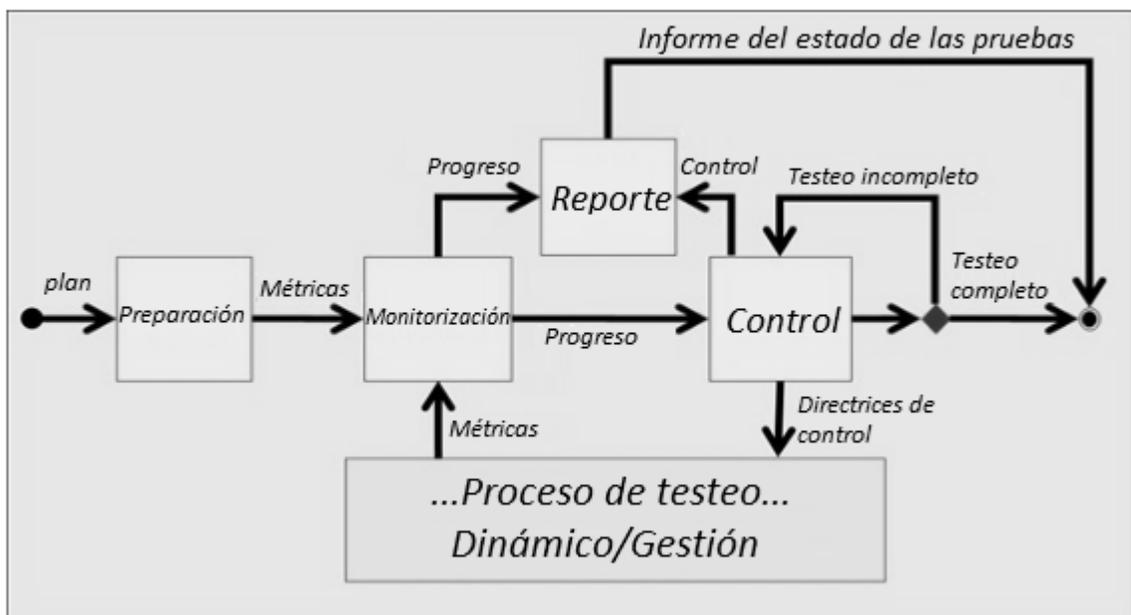


Gráfico 1 Modelo Procesos de Monitoreo y Control de Pruebas

Gráfico 1. Modelo Procesos de Monitoreo y Control de Pruebas. (“IEEE SA - 29119-2-2013 - Software and systems engineering —Software testing —Part 2: Test processes,” 2013) (No es una copia exacta del modelo encontrado en ISO / IEC / IEEE 29119-2)

ISO / IEC / IEEE 29119-4 tiene como objetivo definir un estándar internacional que abarque técnicas de pruebas para evaluar software, también conocidas como técnicas de diseño de casos de prueba o métodos de prueba. Éstas técnicas se pueden emplear durante el diseño de la prueba y en el proceso en que se emplea dentro del desarrollo del software. La base de ésta norma es el estándar de pruebas de componentes BS-7925-2 (ISO/IEC/IEEE, 2015), El estándar BS-7925-2 tenía como objetivo “permitir la medición y comparación de pruebas realizadas en componentes de software. Esto permitirá a los usuarios de este estándar mejorar directamente la calidad de sus pruebas de software y mejorar la calidad de sus productos de software.” (SIGIST, 2001) “Las técnicas de diseño de prueba que se presentan en esta norma pueden utilizarse para derivar casos de prueba que pueden usarse para recopilar pruebas de que se han cumplido los requisitos de cada sistema bajo prueba y / o que se detectan.” (ISO/IEC/IEEE, 2015)

“Las técnicas de pruebas de software se pueden clasificar en dos grandes grupos, pruebas basadas en especificación también conocidas como pruebas de caja negra y pruebas basadas en la estructura también conocidas como pruebas de caja blanca. La diferencia entre las dos radica en que las pruebas de caja blanca se tienen en cuenta la estructura interna del elemento a revisar y en las pruebas de caja negra se realizan en función de entradas y salidas de un proceso en específico.” (Armando et al., 2015) En el proyecto se harán uso de las pruebas de caja negra.

1.8.1.MARCO HISTÓRICO

Los comienzos de la usabilidad inician en la década del setenta (70), los primeros estudios fueron presentados por psicólogos, los cuales estaban interesados en un diseño de sistemas de comunicación o de dialogo entre humanos y computadoras, el objetivo era hacer más eficientes los diálogos, analizando todos los contextos y también el procesamiento de la información para resolver problemas por parte de los usuarios. (Chipantiza, Lewis, Olivo, & Calva, 2015)

En la década de los ochenta (80) se profundizó más en la investigación del estudio sistémico de comunicación hombre máquina. En 1997 es lanzada una norma internacional llamada "Din En ISO 9241-11" que define a la usabilidad como: "la medida de un sistema que puede ser utilizado por usuarios específicos en un contexto específico." (Chipantiza et al., 2015)

Con el transcurso de los años la usabilidad mejoraba constante y se iba adaptando a los cambios del ambiente, aunque por parte de (Morville, 2004) afirmo en el 2004 que la usabilidad es muy necesaria pero insuficiente en el entorno actual, el público que hace uso de los sistemas de información era cada vez más amplio, y en cuestión de manejo, los usuarios finales carecían de ello. (Virgilio, Padilla, Jimenez, & Martin, 2013)

A comienzos del año 2003 el instituto nacional de estándares y tecnología (National Institute of Standards and Technology "NIST") registra más de 30 técnicas para construir casos de usabilidad. A inicios de 2009, la norma (ISO/IEC.2009) define que la usabilidad pasa a ser considerada un atributo de la calidad del software. (Florian, Solarte, & Reyes, 2010). En el primer semestre del año 2017, la usabilidad ya no se centra en sólo facilitar la comunicación entre el humano común y el software, se ha propuesto en el artículo (Alvarez, Jaramillo, & Builes, 2017) la implementación de métodos heurísticos para adaptar la usabilidad del software a personas que presenten alguna discapacidad motriz. La dependencia y el uso de los computadores crece constantemente, volviéndose parte de la vida cotidiana, tanto así que se ve presente en diversas actividad u espacios, como el laboral, entretenimiento, educación e información. Pese a la importancia de dichas actividades, las personas que presentan alguna discapacidad motriz se le dificulta bastante ser participantes de estas. "El método que se propone en este trabajo difiere de los métodos existentes en que toma en cuenta las preferencias de los usuarios y, con base en esto, se categorizan mejor las heurísticas, debido a que los expertos no conocen las necesidades específicas de las personas. La posibilidad de que el usuario final asigne pesos a las características heurísticas, indicando cuáles son las más importantes para ellos y no que el experto fije pesos sin conocer los requisitos de estas personas es lo innovador del método propuesto".

II. DOCUMENTACIÓN DEL SOFTWARE

2.1. PLAN DE PROYECTO

En este segmento del libro, es necesario realizar un plan de proyecto que tenga como finalidad dar respuesta a los objetivos específicos y generales. El proceso consistirá en llevar a cabo una serie de pasos que determinarán las estimaciones que se tienen para dichos objetivos.

- Paso 1: los objetivos deben ser analizados a profundidad por el grupo de trabajo, en este proceso intervienen todos los implicados en el desarrollo del software. Posteriormente se deben tomar decisiones con respecto al tiempo y a la metodología a usar.
- Paso 2: el hardware a usar en el desarrollo es algo fundamental, por esa razón, al finalizar el paso uno se debe analizar y proyectar los recursos técnicos que se requieran para llevar a cabo el proyecto en su totalidad.
- Paso 3: se aconseja que el proceso de desarrollo sea realizado con base a una metodología de desarrollo, valga la redundancia, de esta forma se lleva un correcto flujo de trabajo, realizando procesos en márgenes de tiempo, lo que genera, que se pueda calcular de forma precisa el tiempo de desarrollo.

- Paso 4: La única forma de verificar si los objetivos se cumplieron en el transcurso del proyecto, es mediante resultados. Los resultados indispensables en el proyecto, por esa razón se debe hacer unos de una metodología de investigación, capaz de interpretar los datos que se obtengan del modelo y así demostrar mediante estadística el impacto que genero el desarrollo.

(ver en anexos el calendario de la elaboración del proyecto)

2.2. DETERMINACIÓN DE REQUERIMIENTOS

Cantidad de actividades:	12
Actividad # 1	
Nombre de la actividad	Descripción
Mejoramiento visual de CALISOFT	Esta actividad se mejorará la interfaz gráfica del usuario, creando una mejor experiencia de uso integrando dinamismo a la página.
Actividad # 2	
Nombre de la actividad	Descripción
Aplicar el estándar gráfico que usa el CIT Facatativá (Centro de investigación tecnológico)	Esta actividad se adecuará la interfaz gráfica a la plantilla pre-establecida previamente por el centro de investigación y tecnología.
Actividad # 3	
Nombre de la actividad	Descripción
Hacer validaciones de formularios más rigurosas.	Esta actividad se pretende se le dará más seguridad e integridad al aplicativo web CALISOFT, según las posibles entradas de información por parte del usuario.
Actividad # 4	
Nombre de la actividad	Descripción
Implementar estándar de nomenclatura de base de datos	Esta actividad se implementará el estándar de nomenclatura de base de datos al modelo de CALISOFT.
Actividad # 5	
Nombre de la actividad	Descripción
Implementar estándar de codificación	Esta actividad se implementará el estándar de codificación al código fuente de CALISOFT.
Actividad # 6	
Nombre de la actividad	Descripción
Optimizar el diseño de administración de	En esta actividad se compactará las transacciones que se vean involucradas en los diferentes módulos

recursos.	integrándolas en una sola vista, de esta forma el usuario tendrá que navegar menos para administrar un recurso de la plataforma CALISOFT.
Actividad # 7	
Nombre de la actividad	Descripción
Crear tablas en la base de datos que ayuden en la complementación del módulo de pruebas.	En esta actividad se creará una tabla en la base de datos donde se almacenará la información necesaria para realizar pruebas programadas en el módulo de Testing (Módulo de pruebas).
Actividad # 8	
Nombre de la actividad	Descripción
Facilitar la labor de pruebas funcionales.	Esta actividad se programará la labor en la que el evaluador ingresa los datos en los campos de entrada del formulario en favor de que la plataforma CALISOFT los proporcione.
Actividad # 9	
Nombre de la actividad	Descripción
Integración del módulo de estándares de código fuente.	En esta actividad se integrará el módulo de evaluación de estándar del código fuente al proceso de calificación de la plataforma CALISOFT.
Actividad # 10	
Nombre de la actividad	Descripción
Integración del módulo de estándar de nomenclatura de base de datos.	En esta actividad se integrará el módulo de evaluación de estándar de nomenclatura de base de datos al proceso de calificación de la plataforma CALISOFT.
Actividad # 11	
Nombre de la actividad	Descripción
Evaluación de experiencia de usuario.	En esta actividad será una encuesta a los usuarios que han usado la plataforma previamente sobre la experiencia de usuario con respecto a la nueva interfaz gráfica de la plataforma CALISOFT.
Actividad # 12	
Nombre de la actividad	Descripción
Evaluación de experiencia de usuario de los proyectos evaluados.	En esta actividad se integrará una sección en la parte final del proceso de evaluación donde el evaluador comentará la experiencia interactiva del producto de software previamente calificado.

Tabla 1 Requerimientos del sistema

2.3. ESPECIFICACIÓN DEL DISEÑO

El sistema consiste en evaluar los campos de entrada de información de un producto software, con la finalidad de certificar la calidad y usabilidad de dicho producto. El módulo de Testing cuenta con la intervención de dos actores, un evaluador y un desarrollador. Como paso inicial, el desarrollador debe subir toda la documentación de su proyecto, dentro de la documentación se encontrarán los diagramas de casos de uso, con base a los diagramas de casos de uso, el evaluador procede a crear casos de prueba donde vea que hay un ingreso de datos al software del desarrollador. Con los casos de prueba creados, el desarrollador tiene como objetivo dirigirse al formulario que corresponda y crear una copia del formulario por medio de la extensión de Calisoft, cabe resaltar que la copia se almacena como un JSON. Cuando el caso de prueba es actualizado con el formulario del desarrollador, el evaluador procede a realizar el testeo no heurístico, el sistema realiza una interpretación de los inputs del formulario y únicamente del evaluador debe presionar un botón para que el sistema ingrese datos que obtiene desde la base de datos, cada sentencia que viene de la base de datos tiene un atributo que se llama "valido", de tipo booleano donde un 0 representa que la sentencia corresponde a un dato incoherente o malicioso, y un 1 representa que la sentencia está bien digitada para el input sobre el cual se va a probar y además no es una sentencia maliciosa para el sistema. Resalto que uso el término "Malicioso" para hacer referencia a una sentencia destinada a ataques de Sql o Xss que pueden averiar el funcionamiento del software si se le permite ingreso a la base de datos.

Por el lado del input que el sistema recrea, este viene acompañado de una serie de reglas, que son las mismas que el desarrollador cuenta en su aplicativo web, estas reglas se combinan con un componente de javascript lo cual hace que recree esas mismas reglas sobre el input y que además si se dispara una de ellas, aparecerá un mensaje en color rojo debajo el input. El sistema cuenta con una programación booleana que al momento de ingresar un dato y si el sistema no muestra ningún error por medio del componente de javascript, almacenará un 0, en cambio, si el componente se dispara y muestra un error, el sistema almacenará un 1.

El método de calificación es el siguiente; se emplea una lógica simbólica denominada como "Disyunción exclusiva y operador OR exclusivo (XOR)", se denota por el símbolo \oplus , donde un valor es verdadero cuando dos preposiciones q y p son contrarias y si q y p son dos preposiciones iguales, el valor obtenido es falso (Marcial, Carmona, & Milenio, n.d.).

Sentencia traída de base de datos (p)	Error proveniente del componente de Javascript (q)	Calificación (p \oplus q)
0	1	1
1	0	1
0	0	0
1	1	0

Tabla 2 Disyunción exclusiva XOR

Si la sentencia proveniente de la base de datos (p) es correcta, se denota por un 1, este valor está almacenado en el campo "valido" que anteriormente se mencionó, si no se dispara ningún error al ingresar ese dato, el sistema registrará un 0 para ese error, que significa que no encontró un error en la sentencia para ese tipo de input, implementando XOR obtenemos como resultado un 1, que equivale a una calificación correcta. Por cada inserción de la base de datos sobre un input se aplica la lógica XOR para determinar si está bien o mal las validaciones que uso el desarrollador. Cada prueba tiene una calificación sobre el formulario, la calificación

va ligada a la cantidad de inputs que el formulario posea, se toma un valor de 100% y se divide en la cantidad de inputs, eso le dará a cada input un porcentaje, por ejemplo, si un formulario tiene cuatro inputs, cada input tendrá un valor de 25%, si la calificación obtenida por la lógica de XOR es positiva, el valor se cuenta como un 25 entero, pero si es negativa, la calificación se toma como 0, al final se suman todos los resultados como se aprecia en la formula (1).

(1)

$$CalificacionParcial = \sum_{i=0}^{CantidadInputs} \frac{100}{CantidadInputs} \times ResultadoXOR$$

Una vez obtenida la calificación parcial de cada prueba, se procede a realizar una sumatoria de cada calificación parcial y dividirla sobre el número total de calificaciones parciales que se hicieron un formulario en específico, de ese modo, se obtiene la calificación final para dicho formulario, este proceso corresponde a la formula (2).

(2)

$$CalificacionFinal = \frac{\sum CalificacionParcial}{CantidadDeCalificacionesParciales}$$

El valor obtenido es guardado en la tabla donde está creado el caso prueba del correspondiente formulario en el apartado de “calificación”.

2.3.1. MODELO DE IDENTIDAD RELACIÓN (MER)

“El diagrama MER es un modelo de red, que describe con un alto nivel de abstracción, la distribución de datos almacenados en un sistema. Los diagramas ER se centran en los datos y en sus interrelaciones y por ello, no representan la estructura para el modelado de otros elementos del proceso. Dichos diagramas son representaciones completamente estáticas y no proporcionan la información en el tiempo para poder analizarla y medirla”. (Sanchis, Poler, & Ortiz, 2009)

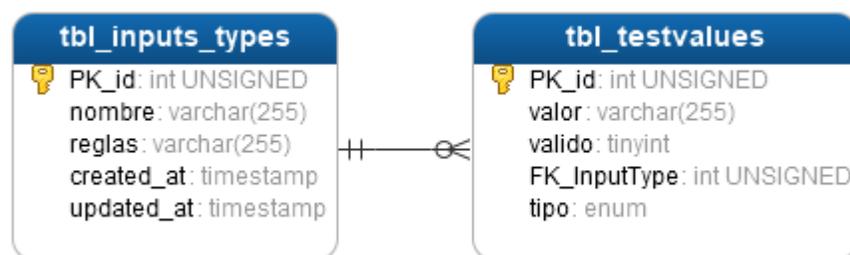


Gráfico 2 Diagrama de entidad relación (MER)

Tabla	Descripción
Tbl_inputs_types	En esta tabla se almacenan los diferentes tipos de inputs que puede haber en un formulario, por ejemplo, un tipo email, contraseña, text, number, etc. Entre sus campos se puede guardar el nombre del tipo, las reglas con las que se valida la información y los datos que ingresan mediante ese tipo de inputs y dos campos que almacenan la fecha en que se crea y la fecha en que se realice una modificación desde el sistema.
Tbl_testvalues	Esta tabla almacena los diferentes valores con los que trabajará el proceso de Testing no heurístico del módulo de plataforma, en el campo de valor se asignará una sentencia que corresponda al tipo de input en el que se utilizará, por ejemplo, para el tipo email, en valor se almacenará una sentencia como; prueba@email.com o pruebaemail.com, como se puede observar, una sentencia es correcta y la otra es incorrecta, ese valor de incorrecto e incorrecto se almacena en el campo de la tabla "valido", donde 1 será correcto y 0 incorrecto. La columna FK_InputType corresponde a la llave foránea de la tabla Tbl:inputs:types, y el campo llamado "tipo" corresponde a al tipo de ataque, si es html, xss o sql.

Tabla 3 Descripción Modelo Entidad Relación

2.3.2. DIAGRAMAS DE CASOS DE USO

“Un caso de uso puede ser definido como una secuencia de acciones, incluyendo variaciones, que el sistema puede ejecutar y que produce un resultado observable de valor para un actor que interactúa con el sistema” (García, Ortín, Moros, Nicolás, & Tova, 2007)

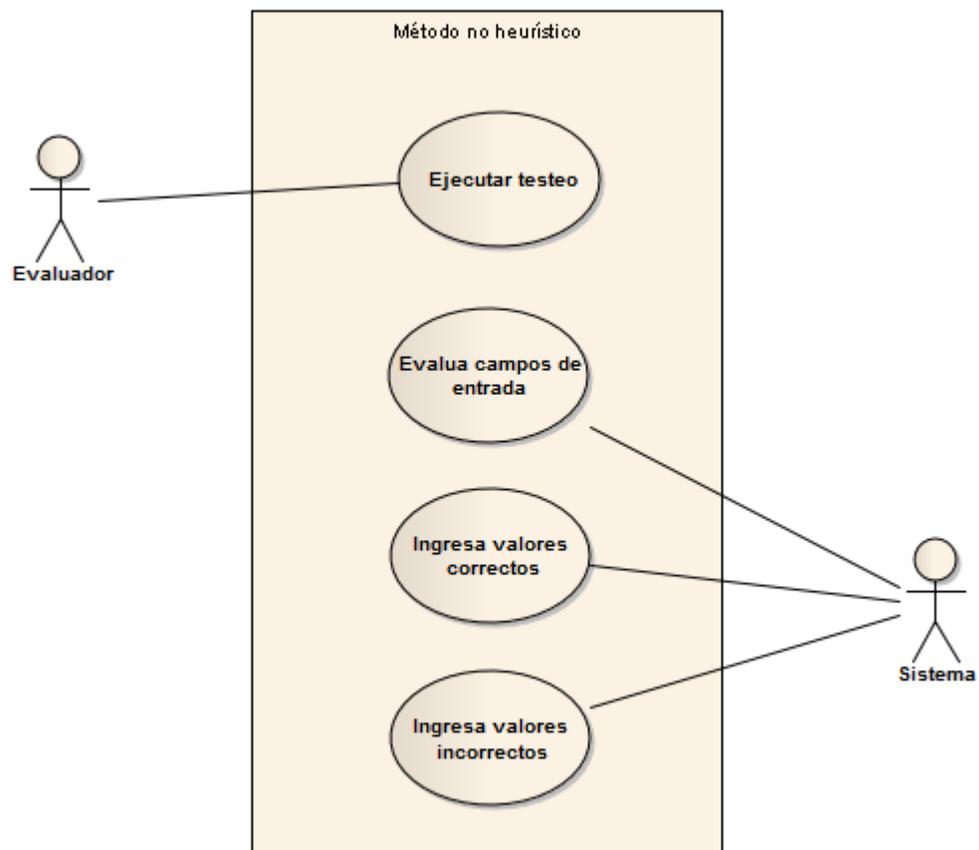


Gráfico 3 Diagrama de casos de uso

DESCRIPCIÓN DE CASOS DE USO

ROLES

ROL	DESCRIPCIÓN
Evaluador:	Ejecución de las diferentes pruebas a los formularios alojados en su módulo de calificación
Sistema:	Lógica de testeo no heurístico para la evaluación entre los diferentes tipos de entradas de información.

Tabla 4 Descripción Roles, casos de uso

CASOS DE USO

Nombre:	Ejecutar Testeo
Alias:	
Actores:	Evaluador
Función:	Inicia la función de Testeo sobre el formulario
Descripción:	El evaluador analizará los diferentes campos de entrada y ejecutará el script que realizará el testeo
Referencias:	

Tabla 5 Ejecutar testeo, casos de uso

Nombre:	Evalúa Campos de Entrada
Alias:	
Actores:	Sistema
Función:	Analiza los inputs del formulario
Descripción:	Revisa los diferentes inputs con sus diferentes tipos de entrada y procede a comprenderlos y agruparlos.
Referencias:	

Tabla 6 Evalúa campos de entrada, casos de uso

Nombre:	Ingresar Valores Correctos
Alias:	
Actores:	Sistema
Función:	Ingresar valores correctos para el input
Descripción:	Según una base de datos preestablecida consulta e inserta los valores correctos para ese tipo de input
Referencias:	

Tabla 7 Ingresar valores correctos, casos de uso

Nombre:	Ingresar Valores Incorrectos
Alias:	
Actores:	Sistema
Función:	Ingresar valores correctos para el input
Descripción:	Según una base de datos preestablecida consulta e inserta los valores incorrectos para ese tipo de input.
Referencias:	

Tabla 8 Ingresar valores correctos, casos de uso

2.3.3. DIAGRAMAS DE SECUENCIAS

El diagrama de secuencias describe el comportamiento de actores y usuarios dentro del sistema, el actor se muestra con el símbolo de una persona y los objetos como una caja en la parte superior de una línea vertical punteada. La línea vertical punteada corresponde a la línea de vida de cada objeto o actor durante el ciclo de la operación. Los mensajes que comunican cada objeto o usuarios entre ellos, se representan con una flecha entre sus líneas de vida. El flujo del diagrama se debe leer con forme al flujo de los mensajes. (FOWLER, 1999)

SECUENCIA EJECUTAR TESTEO

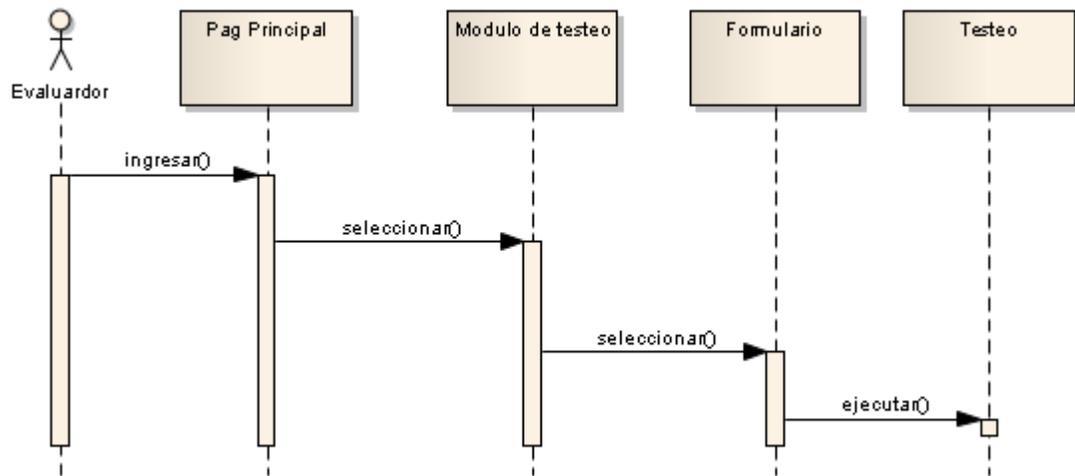


Gráfico 4 Diagrama de secuencia, ejecutar testeo

Tabla	Descripción
Evaluador	Corresponde al actor, en este caso es el desarrollador, encargado de realizar el caso prueba de dar inicio al proceso de testeo.
Pag principal	Es la página principal del evaluador, por así decirlo, su dashboard, e donde las opciones necesarias para dirigirse a la sección donde se encuentra el módulo de plataforma, que corresponde al módulo de Testing.
Módulo de testeo	Dentro del módulo de plataforma, encontrará los casos prueba que haya subido el desarrollador, el desarrollador deberá escoger el que desea evaluar.
Formulario	Cada caso de uso se entiende como un formulario, el sistema almacena copias de los formularios que envía el desarrollador mediante el aplicativo.
Testeo	Es una acción que se ejecuta sobre un formulario, realiza el ingreso dinámico de datos sobre todos los inputs que contenga la copia del formulario que subió el desarrollador. Este ingreso dinámico y evalúa que los campos cuenten con la seguridad para evitar el ingreso de datos erróneos o maliciosos.

Tabla 9 Descripción secuencia ejecutar testeo

SECUENCIA EVALÚA CAMPOS DE ENTRADA

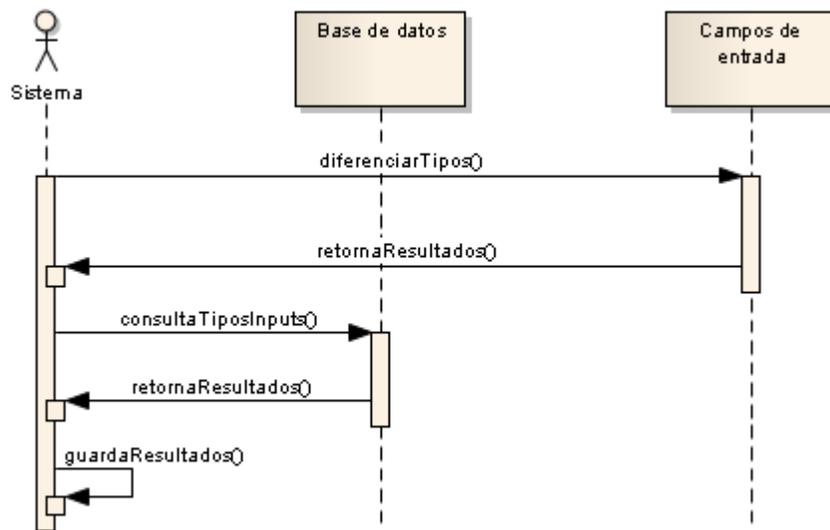


Gráfico 5 Diagrama de secuencia, evalúa campos de entrada

Tabla	Descripción
Sistema	Corresponde a un actor, que es el sistema lógico del aplicativo, encargado interpretar componentes en el aplicativo y de traer información de la base de datos para realizar las pruebas de Testing.
Base de datos	La base de datos contiene las sentencias que será usadas por cada tipo de input que encuentra en un formulario.
Campos de entrada	Corresponde a los inputs que se hayan en la copia del formulario que sube un desarrollador, el sistema analizará los campos de entrada y retornará un resultado que es el tipo de ese input.

Tabla 10 Descripción secuencia evalúa campos de entrada

SECUENCIA INGRESA VALORES CORRECTOS

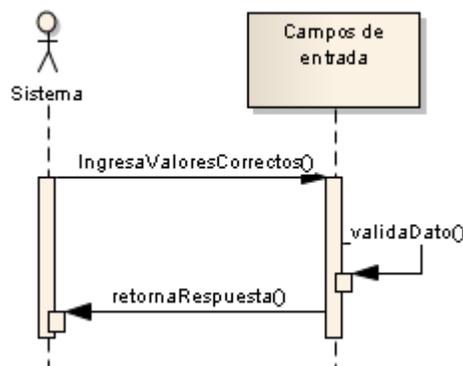


Gráfico 6 Diagrama de secuencia, ingresar valores correctos

Tabla	Descripción
Sistema	Corresponde a un actor, que es el sistema lógico del aplicativo, encargado interpretar componentes en el aplicativo y de traer información de la base de datos para realizar las pruebas de Testing.
Campos de entrada	El sistema guarda los valores necesarios para usarse en este paso. Ingresar los valores únicamente correctos de forma automática y dinámica, el input comprueba las validaciones en tiempo real y da un resultado positivo o negativo al sistema.

Tabla 11 Descripción secuencia ingresa valores correctos

SECUENCIA INGRESA VALORES INCORRECTOS

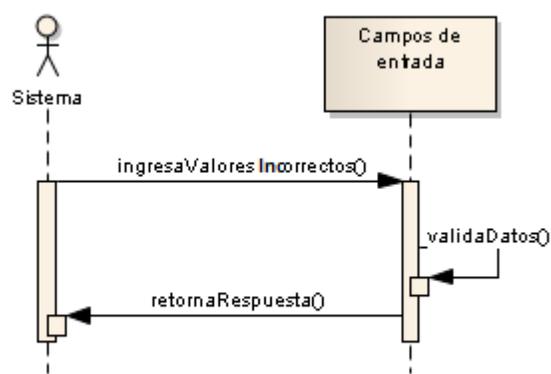


Gráfico 7 Diagrama de secuencia, ingresa valores incorrectos

Tabla	Descripción
Sistema	Corresponde a un actor, que es el sistema lógico del aplicativo, encargado interpretar componentes en el aplicativo y de traer información de la base de datos para realizar las pruebas de Testing.
Campos de entrada	El sistema guarda los valores necesarios para usarse en este paso. Ingresar los valores únicamente incorrectos de forma automática y dinámica, el input comprueba las validaciones en tiempo real y da un resultado positivo o negativo al sistema.

Tabla 12 Descripción ingresa valores incorrectos

SECUENCIA CENTRAL

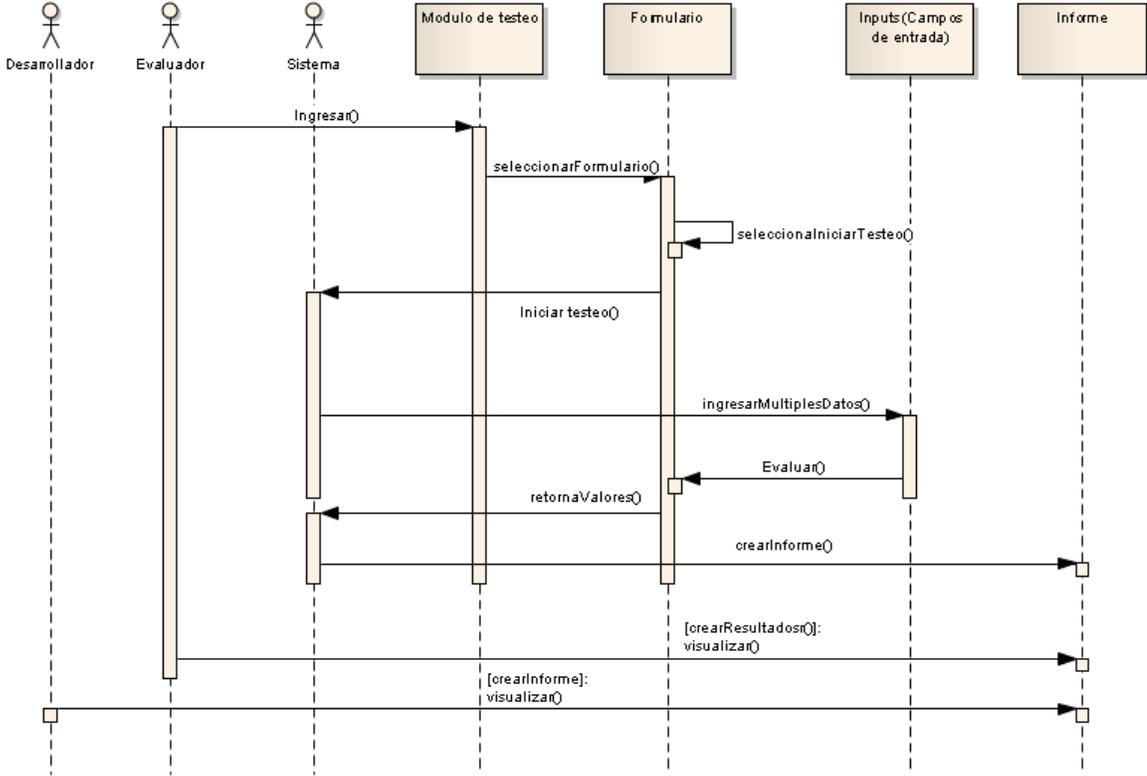


Gráfico 8 Diagrama de secuencia, central

Tabla	Descripción
Desarrollador	Corresponde al actor que se encarga de subir los formularios correspondientes a los casos de prueba que se hagan sobre su aplicativo, este usuario al final del proceso de Testing, podrá ver los resultados en su dashboard, estos resultados serán mostrados por medio de un informe.
Evaluador	Corresponde al actor, en este caso es el desarrollador, encargado de realizar el caso prueba de dar inicio al proceso de testeo.
Sistema	Corresponde a un actor, que es el sistema lógico del aplicativo, encargado de interpretar componentes en el aplicativo y de traer información de la base de datos para realizar las pruebas de Testing.
Módulo de testeo	Dentro del módulo de plataforma, encontrará los casos prueba que haya subido el desarrollador, el desarrollador deberá escoger el que desea evaluar.
Formulario	Cada caso de uso se entiende como un formulario, el sistema almacena copias de los formularios que envía el desarrollador mediante el aplicativo.
Inputs (campos de entrada)	El sistema guarda los valores necesarios para usarse en este paso. Ingresan los valores correctos e incorrectos de forma automática y dinámica, el input comprueba las validaciones en tiempo real y da un resultado positivo o negativo al sistema.
Informe	El informe contiene todos los resultados y valores con los que se realizó la prueba de testeo. Estará a disposición tanto del evaluador como del desarrollador.

Tabla 13 descripción secuencia general

2.3.4. DIAGRAMA DE ACTIVIDADES

Los diagramas de actividades explican el orden en que se realizan ciertas tareas dentro del sistema, describiendo un flujo que bien puede ser paralelo o secuencial que puede llevar a varios finales, el final se interpreta como la conclusión del objetivo de la actividad. (FOWLER, 1999)

ACTIVIDAD EJECUTAR TESTEO

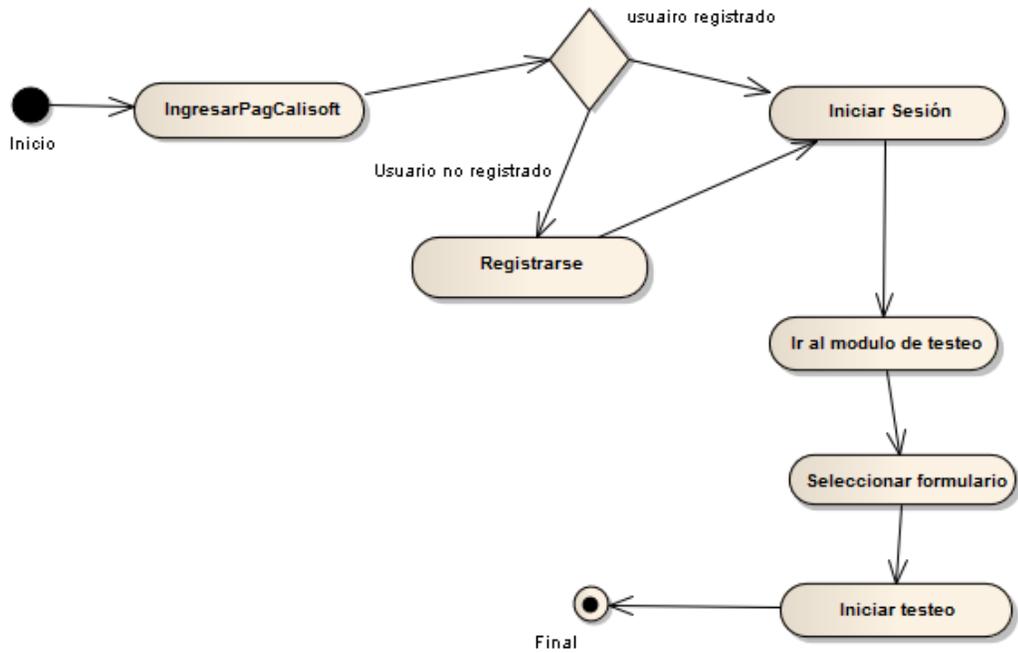


Gráfico 9 Diagrama de actividades, ejecutar testeo

Tabla	Descripción
IngresarPagCalisoft	Ingresar a la página principal de Calisoft donde se encuentran las opciones de inicio de sesión y registro.
Iniciar Sesión	Si es un usuario existente en la base de datos, procederá a digitar su correo y contraseña para iniciar sesión en la plataforma.
Registrarse	Si el usuario no se a registrado, debe proceder a registrarse mediante la página principal para poder ingresar a la plataforma con el rol de desarrollador o contactar al administrador de Calisoft para que cree un usuario con un rol de evaluador.
Ir al módulo de testeo	Sección donde se encuentran los diferentes casos pruebas que almacenan los formularios que ha subido un desarrollador.
Seleccionar formulario	Seleccionar un caso prueba para acceder a el formulario deseado.
Iniciar testeo	Ejecutar el testeo sobre el formulario deseado. El sistema iniciará el proceso de evaluación.

Tabla 14 Descripción actividad ejecutar testeo

ACTIVIDAD EVALÚA CAMPOS DE ENTRADA

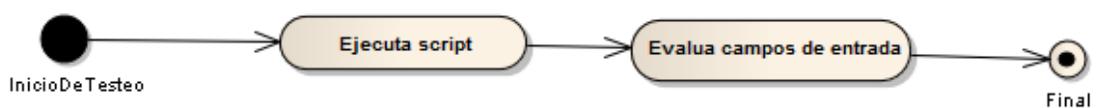


Gráfico 10 Diagrama de actividades, evaluar campos de entrada

Tabla	Descripción
Ejecutar script	Ejecutar el script que da inició al proceso de testeó sobre un formulario.
Evalúa campos de entrada	El sistema evalúa los diferentes inputs que contenga el formulario que subió el desarrollar.

Tabla 15 Descripción evalúa campos de entrada

ACTIVIDAD INGRESA VALORES CORRECTOS

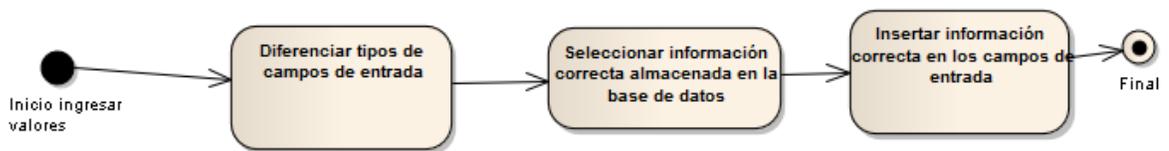


Gráfico 11 Diagrama de actividades, ingresar valores correctos

Tabla	Descripción
Diferenciar tipos de campos de entrada	El sistema almacena la información sobre el tipo de inputs que contiene el formulario.
Seleccionar información correcta almacenada en la base de datos	El sistema selecciona la información necesaria y correcta que se encuentra almacenada en la base de datos, esto es para cada tipo de input.
Insertar información correcta en los campos de entrada.	Según el tipo de input, el sistema procederá a insertar las sentencias obtenidas de la base de datos etiquetadas como correctas, para así realizar el ingreso automático de los valores sobre el formulario.

Tabla 16 Descripción actividad ingresa valores correctos

ACTIVIDAD INGRESA VALORES INCORRECTOS

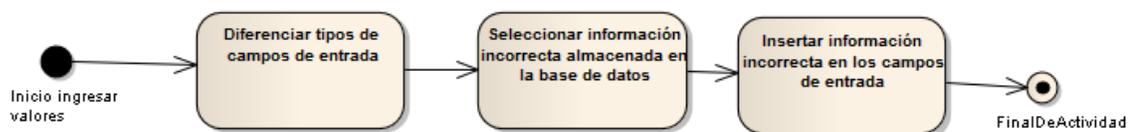


Gráfico 12 Diagrama de actividades, ingresar valores incorrectos

Tabla	Descripción
Diferenciar tipos de campos de entrada	El sistema almacena la información sobre el tipo de inputs que contiene el formulario.
Seleccionar información correcta almacenada en la base de datos	El sistema selecciona la información necesaria y correcta que se encuentra almacenada en la base de datos, esto es para cada tipo de input.
Insertar información correcta en los campos de entrada.	Según el tipo de input, el sistema procederá a insertar las sentencias obtenidas de la base de datos etiquetadas como correctas, para así realizar el ingreso automático de los valores sobre el formulario.

Tabla 17 Descripción actividad ingresa valores incorrectos

ACTIVIDADES CENTRALES

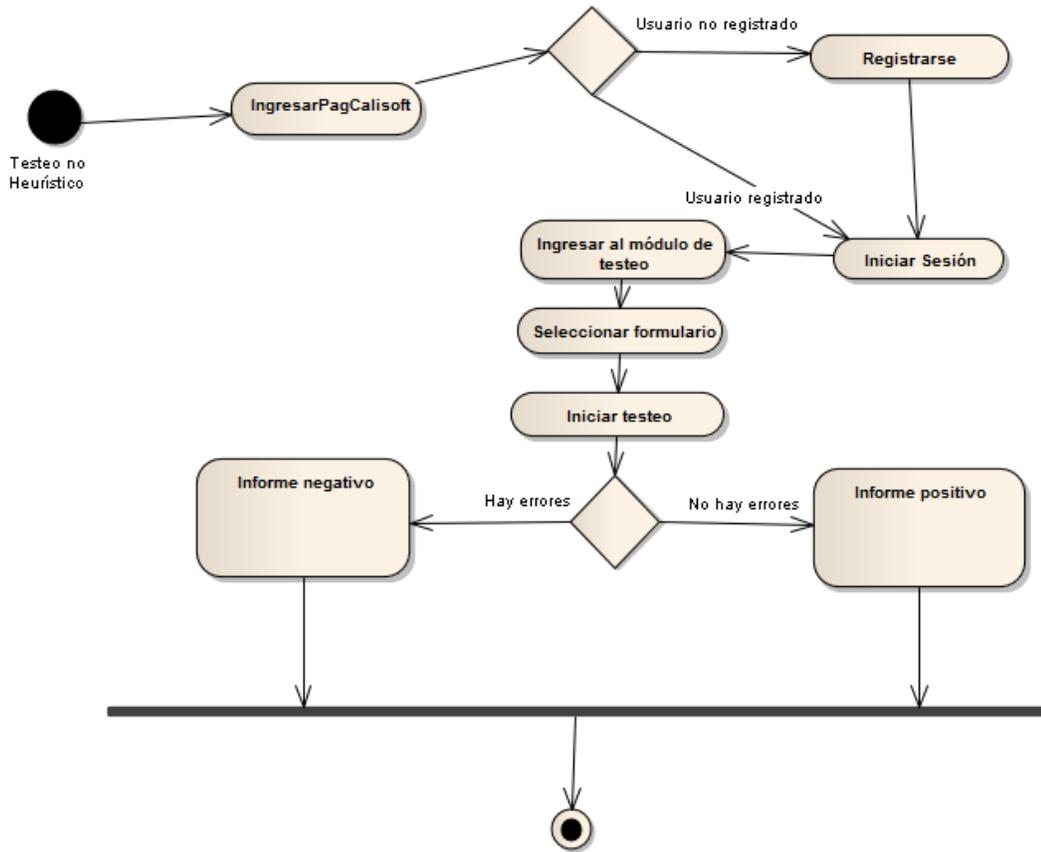


Gráfico 13 Diagrama de actividades, general

Tabla	Descripción
IngresarPagCalisoft	Ingresar a la página principal de Calisoft donde se encuentran las opciones de inicio de sesión y registro.
Iniciar Sesión	Si es un usuario existente en la base de datos, procederá a digitar su correo y contraseña para iniciar sesión en la plataforma.
Registrarse	Si el usuario no se a registrado, debe proceder a registrarse mediante la página principal para poder ingresar a la plataforma con el rol de desarrollador o contactar al administrador de Calisoft para que cree un usuario con un rol de evaluador.
Ingresar al módulo de testeo	Sección donde se encuentran los diferentes casos pruebas que almacenan los formularios que ha subido un desarrollador.
Seleccionar formulario	Seleccionar un caso prueba para acceder a el formulario deseado.
Iniciar testeo	Ejecutar el testeo sobre el formulario deseado. El sistema iniciará el proceso de evaluación.
Informe negativo	Según el resultado que el sistema haya obtenido con base a las sentencias probadas sobre los inputs del formulario que subió el desarrollador, el sistema creará un formulario con una nota negativa, donde se le informará al desarrollador todas las sentencias que se probaron en los inputs de su formulario.
Informe positivo	Según el resultado que el sistema haya obtenido con base a las sentencias probadas sobre los inputs del formulario que subió el desarrollador, el sistema creará un formulario con una nota positiva, donde se le informará al desarrollador todas las sentencias que se probaron en los inputs de su formulario.

Tabla 18 Descripción actividades centrales

2.3.5. DIAGRAMA DE CLASES

Los diagramas de clases son excelentes para capturar el lenguaje del sistema. “El diagrama de clase describe los tipos de objetos que hay en el sistema y las diversas clases de relaciones estáticas que existen entre ellos”. (FOWLER, 1999) También describen todos los atributos y operaciones que posea una clase, además de las restricciones en las que se encuentren sujetos, esto depende de la forma en la que se conecten entre ellas.

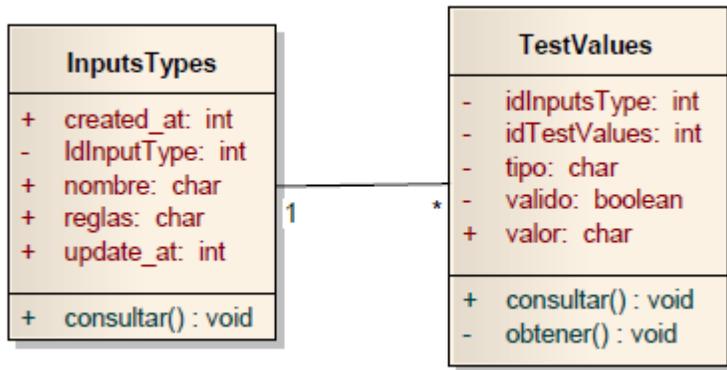


Gráfico 14 Diagrama de clases

Tabla	Descripción
InputsTypes	Contiene los tipos de inputs que se pueden hallar en un formulario html, además de las reglas que regularán un estándar para el procedimiento de evaluación.
Testvalues	Contiene una variable que lo relaciona con el tipo de input, además contiene un valor que representa una sentencia incorrecta o correcta para cada tipo de input, si es correcta o no se debe estipular en el campo "Valido" de tipo booleano. El campo "tipo" define si la sentencia corresponde a un ataque Xss. Sql o Html.

Tabla 19 Descripción diagrama de clases

2.3.6. DIAGRAMA DE COLABORACIÓN

En los diagramas de colaboración, los objetos se muestran como cajas y los actores con un símbolo de una persona, la comunicación entre ellos se da por medio de flechas, similar al diagrama de secuencia, pero en este caso, los mensajes están enumerados describiendo el orden en que deben ser interpretados, esos diagramas son útiles para mostrar la vinculación de los elementos dentro del sistema. (FOWLER, 1999)

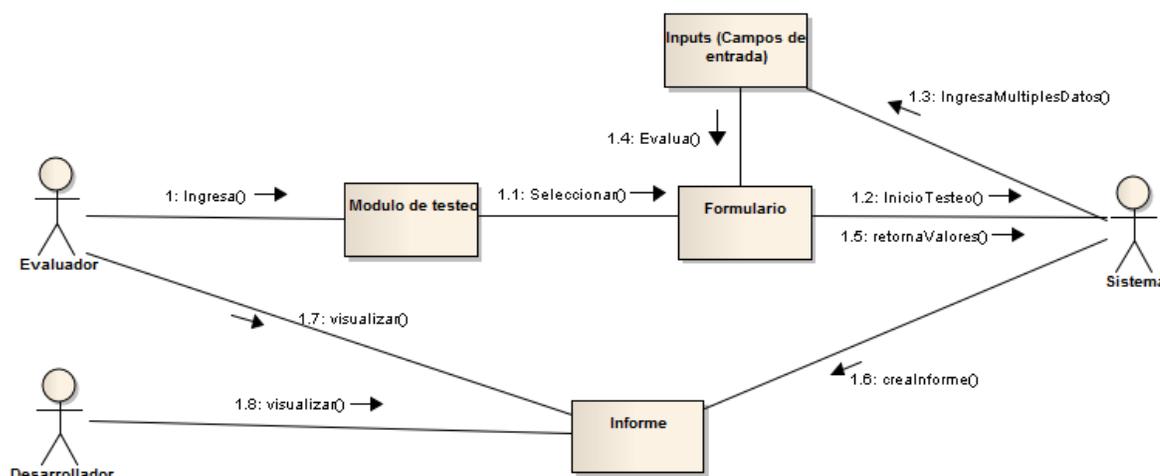


Gráfico 15 Diagrama de colaboración

Tabla	Descripción
Desarrollador	Corresponde al actor que se encarga de subir los formularios correspondientes a los casos de prueba que se hagan sobre su aplicativo, este usuario al final del proceso de Testing, podrá ver los resultados en su dashboard, estos resultados serán mostrados por medio de un informe.
Evaluador	Corresponde al actor, en este caso es el desarrollador, encargado de realizar el caso prueba de dar inicio al proceso de testeo.
Sistema	Corresponde a un actor, que es el sistema lógico del aplicativo, encargado de interpretar componentes en el aplicativo y de traer información de la base de datos para realizar las pruebas de Testing.
Módulo de testeo	Dentro del módulo de plataforma, encontrará los casos prueba que haya subido el desarrollador, el desarrollador deberá escoger el que desea evaluar.
Formulario	Cada caso de uso se entiende como un formulario, el sistema almacena copias de los formularios que envía el desarrollador mediante el aplicativo.
Inputs (campos de entrada)	El sistema guarda los valores necesarios para usarse en este paso. Ingresan los valores correctos e incorrectos de forma automática y dinámica, el input comprueba las validaciones en tiempo real y da un resultado positivo o negativo al sistema.
Informe	El informe contiene todos los resultados y valores con los que se realizó la prueba de testeo. Estarán a disposición tanto del evaluador como del desarrollador.

Tabla 20 Descripción diagrama de colaboración

2.3.7. DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue es muy útil para facilitar la comprensión del modelo entre los ingenieros del hardware y del software, ya que modela la arquitectura en tiempo de ejecución de un sistema, mostrando los elementos del hardware y software empleados. (Ferré & Sánchez, 2002)

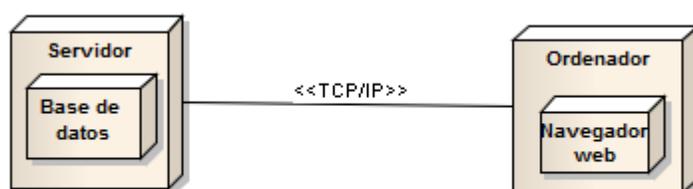


Gráfico 16 Diagrama de despliegue

The image shows a web form titled "Crear Caso Prueba" with a close button (X) in the top right corner. The form is divided into several sections:

- Nombre:** A text input field containing "Prueba Prueba Prue".
- Resultado esperado:** An empty text input field.
- Proposito:** A text input field containing "Prueba Prueba". A red error message is displayed below it: "El campo proposito debe contener 200 caracteres como máximo." This error message is enclosed in a green rectangular box.
- Alcance:** A text input field containing "Prueba Prueba Prue".
- Criterios:** An empty text input field.
- Prioridad:** A dropdown menu with "Alta" selected.
- Límite:** A date input field containing "26/04/2018".

At the bottom of the form, there are two buttons: a green "+ CREAR" button and a red "CANCELAR" button with a white 'X' icon.

Imagen 2 Validación en campos de caso prueba, evaluador

Como se puede observar en las imágenes anteriores, el campo nombre no permite que haya más de 50 caracteres, y los demás campos no se permite que sobrepasen los 200 caracteres, esto permite que no ingresen textos extensos que puedan ocasionar conflicto en la base de datos o en la estética del sistema. También se implementó una validación para el campo de fecha límite, para que el evaluador no escogiera una fecha anterior a la actual, dado que la fecha límite es primordial para que el desarrollador tenga tiempo necesario para subir su formulario.

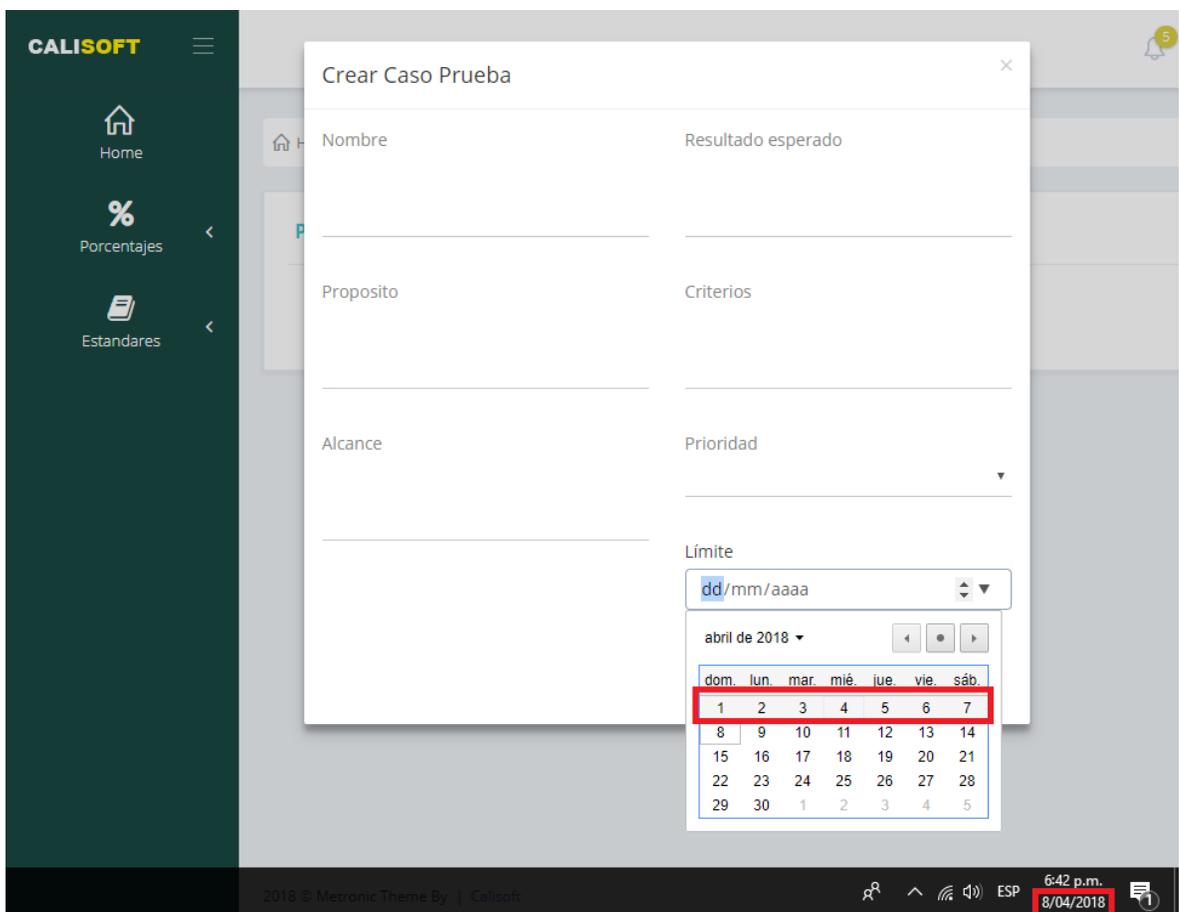


Imagen 3 Validación de fecha límite, evaluador

Del lado del desarrollador, al momento de subir su formulario cuenta únicamente con dos campos de texto, una que corresponde al texto JSON que da la extensión y el segundo que corresponde a una observación.

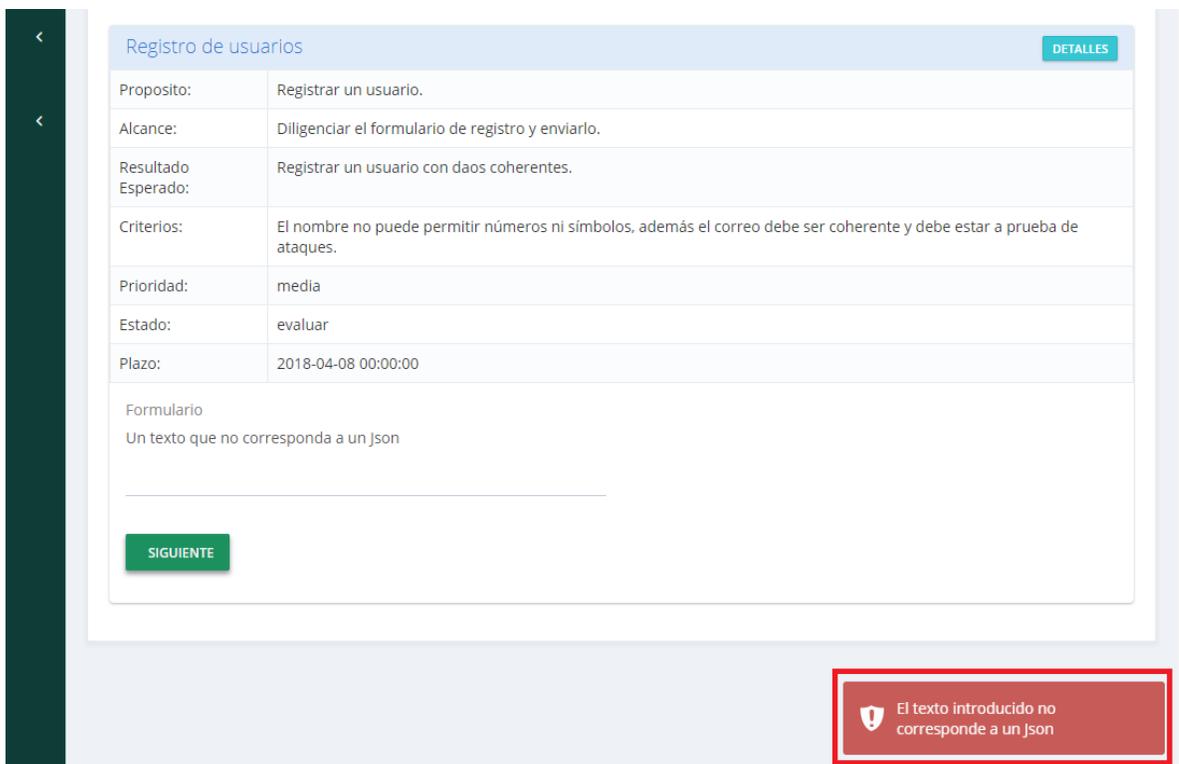


Imagen 4 Validación de JSON, desarrollador

Como se puede observar en la imagen anterior, si se digita algo diferente a un formato tipo JSON, el sistema envía una notificación en la parte inferior derecha de color rojo, que informa de que el texto introducido no corresponde al formato

esperado. Al momento de dar clic en el botón “SIGUIENTE” el sistema abrirá el siguiente modal; en el caso del campo de observación.

The image shows a modal window titled "Enviar Caso Prueba". It contains a text area for "Observación (Opcional)" with a red border and a red error message: "El campo observacion debe contener 150 caracteres como máximo." Below the text area are four dropdown menus labeled "Email", "Nombre persona", "Password", and "Email". To the right of these dropdowns are four red-bordered boxes containing labels: "NOMBRE: EMAIL ID:EMAIL", "NOMBRE: NAME ID:NAME", "NOMBRE: PASSWORD ID:PASSWORD", and "NOMBRE: PASSWORD_CONFIRMATION ID:PA". At the bottom left of the modal is a green button labeled "SUBIR CASO PRUEBA".

Como es un campo opcional, si se deja el blanco el sistema lo deja pasar y automáticamente ingresa al sistema una sentencia que dice “No hay observación” y en el caso de la longitud, tiene un límite de 200 caracteres para evitar el ingreso de información extensa en la base de datos.

Cada caso prueba tiene una fecha límite, si el desarrollador se excede de esa fecha límite inmediatamente el sistema le reporta lo siguiente;

The image shows a screenshot of a web application interface. On the left is a dark green sidebar with icons for "Home", "Evaluación", "Porcentajes", and "Estandares". The main content area has a breadcrumb "Home > PlataformaStudent" and a title "PLATAFORMA: PROYECTO DE PRUEBA". Below the title are sections for "Registro" and "Login", each with a "PRUEBAS DE" button. A section titled "Registro de usuarios" contains a table with the following data:

Proposito:	Registrar un usuario.
Alcance:	Diligenciar el formulario de registro y enviarlo.
Resultado Esperado:	Registrar un usuario con daos coherentes.
Criterios:	El nombre no puede permitir números ni símbolos, además el correo debe ser coherente y debe estar a prueba de ataques.
Prioridad:	media
Estado:	evaluar
Plazo:	2018-04-07 00:00:00

Below the "Plazo" row, there is a red-bordered box containing a red error message: "SE HA PASADO DE LA FECHA LÍMITE PARA SUBIR ESTE CASO PRUEBA."

Imagen 5 Validación de fecha límite

Inhabilitándolo de la opción de subir el formulario, en este caso se debe comunicar con el evaluador para que vuelva a habilitar el caso prueba, pero es una decisión que queda bajo el criterio del evaluador.

Por último, en el proceso de evaluación el funcionamiento es sencillo y eficaz, dado que las pruebas son automatizadas.

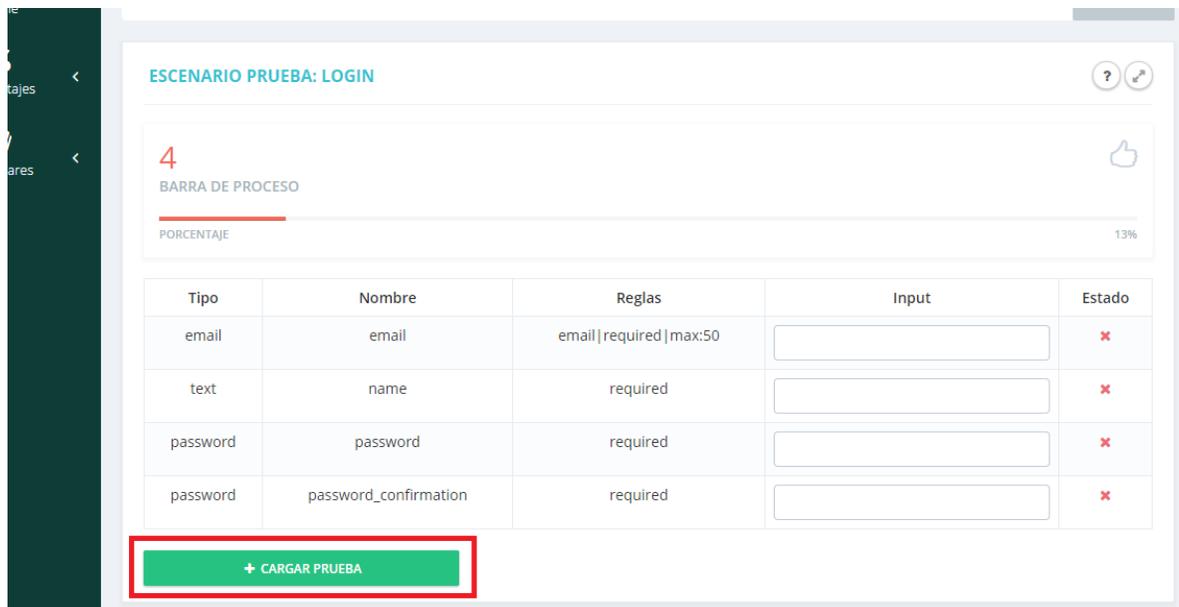


Imagen 6 Funcionamiento de módulo de Testing

Para el correcto funcionamiento es necesario dar clic en el botón “CARGAR PRUEBA” y el sistema automáticamente ingresará valores provenientes de la base de datos llenado los correspondientes inputs del formulario que haya cargado el desarrollador.

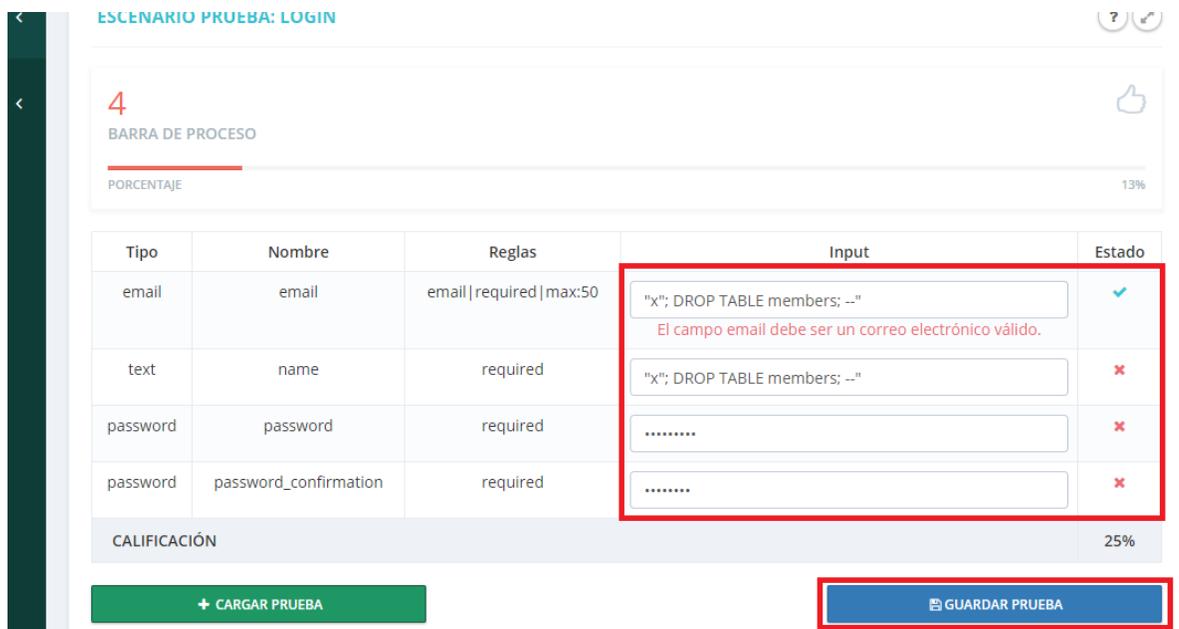


Imagen 7 Validación en tiempo real

En tiempo real los inputs serán evaluados y notificarán si cumplen con las validaciones de datos coherentes y correctos, y si presenta una correcta validación para ataques XSS y ataques SQL. Como se puede apreciar en la imagen anterior, el primer campo cuenta con las validaciones correctas dado que se disparó el error que notifica que el campo debe ser un correo válido, en cambio, en los siguientes campos presenta una mala validación, dado que campo nombre debía evitar valores y caracteres especiales. Al oprimir el botón “CARGAR PRUEBA” es necesario que se oprima el botón “GUARDAR PRUEBA” para que se almacene la prueba.

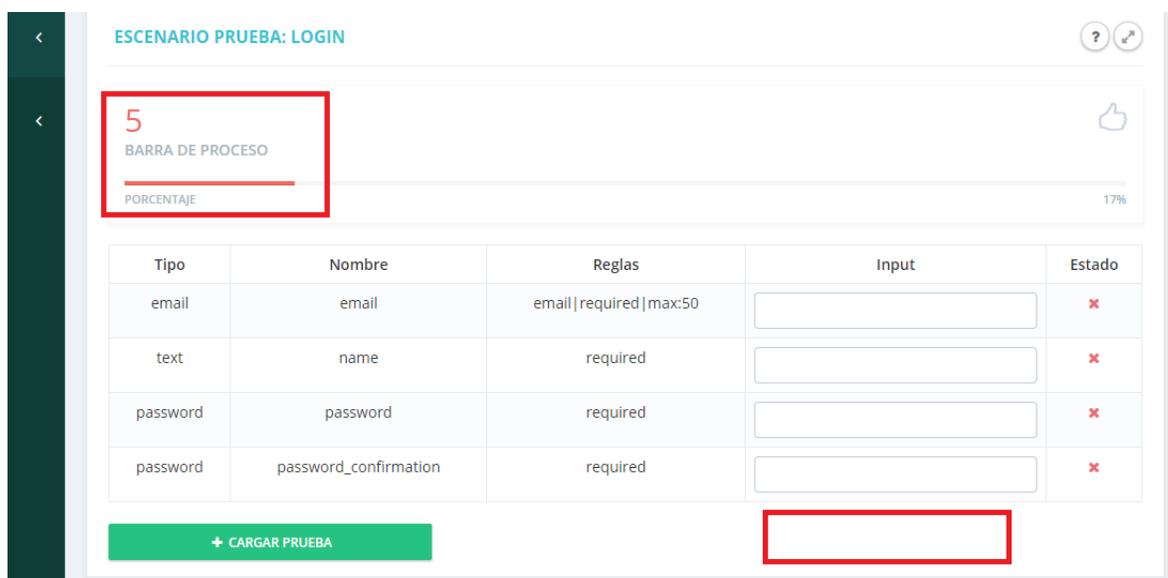


Imagen 8 Prueba guardada

Los campos de los inputs se borran, el botón de guardar prueba desaparece y la barra de proceso crece, este mismo proceso se debe repetir hasta llenar la barra de proceso y con ello se habrá finalizado el proceso de Testing sobre un formulario.

2.5. ESTIMACIÓN DE RECURSOS

FINANCIACIÓN (FUENTES)

TIPO DE FUENTE INTERNA	FUENTE UNIVERSIDAD DE CUNDINAMARCA	VALOR APORTADO (en efectivo y/o especie)
Interna	Universidad de Cundinamarca programa ingeniería de sistemas	40'000.000

Tabla 22 Financiación

RESUMEN POR RUBROS

Rubros	Solicitado en efectivo a UDEC	Contrapartida en especie		Total
		UDEC	Otras Entidades	
PERSONAL	32'000.000			32'000.000
EQUIPOS	4'000.000			4'000.000
MATERIALES E INSUMOS	300.000			300.000
SERVICIOS TECNOLOGICOS	2'000.000			2'000.000
VIAJES	600.000			600.000
OTROS	1'100.000			1'100.000
TOTALES	40'000.000			40'000.000

Tabla 23 Resumen por rubros

DETALLE DE RUBROS

Detalle de personal

Nombre	Función en el proyecto	Tipo de vinculación	Dedicación Horas/semana	Entidad a la que pertenece	Solicitado en efectivo a UDEC	Contrapartida en especie		Total
						UDEC	Otras Entidades	
Cesar Barahona	Investigador principal	Docente	8 H	UDEC	\$16'000.000			\$16'000.000
Johan Suárez	Estudiante pregrado investigador auxiliar	Estudiante	8 H	UDEC	\$8'000.000			\$8'000.000
Stevenson Rangel	Estudiante pregrado investigador auxiliar	Estudiante	8 H	UDEC	\$8'000.000			\$8'000.000

Tabla 24 Detalle de personal

Descripción de equipos

Descripción	Justificación	Cantidad	Valor Unitario	Solicitado en efectivo a UDEC	Contrapartida en especie		Total
					UDEC	Otras Entidades	
Portátil Asus	Herramienta de trabajo	1	Externa	\$2'000.000			\$2'000.000
Portátil Asus	Herramienta de trabajo	1	Externa	\$2'000.000			\$2'000.000

Tabla 25 Descripción de equipos

Descripción de materiales e insumos

Descripción	Justificación	Cantidad	Valor Unitario	Solicitado en efectivo a UDEC	Contrapartida en especie		Total
					UDEC	Otras Entidades	
Papelería	Documentación	N/A	Externa	\$300.000			\$300.000

Tabla 26 Descripción de materiales e insumos

Descripción de servicios tecnológicos

Descripción	Justificación	Valor	Entidad	Solicitado en efectivo a UDEC	Contrapartida en especie		Total
					UDEC	Otras Entidades	
Servidores	Fuente de investigación	\$1'000.000	Externa	\$1'000.000			\$1'000.000
Conexión a Internet	Fuente de investigación	\$1'000.000	Externa	\$1'000.000			\$1'000.000

Tabla 27 Descripción de servicios tecnológicos

Descripción de viajes

Lugar/justificación	No. Días	No. personas	Cantidad	Valor Unitario	Solicitado en efectivo a UDEC	Contrapartida en especie		Total
						UDEC	Otras Entidades	
Conferencias, Capacitaciones	3	3		\$200.000	\$600.000			\$600.000

Tabla 28 Descripción de viajes

Descripción de otros

Descripción	Justificación	Cantidad	Valor Unitario	Solicitado en efectivo a UDEC	Contrapartida en especie		Total
					UDEC	Otras Entidades	
Imprevistos	Daño de equipo	N/A	N/A	\$400.000			\$400.000
Actualización	Continuo mejoramiento	N/A	N/A	\$300.000			\$300.000
Licencias	Adquisición de nuevo software	N/A	N/A	\$400.000			\$400.000

Tabla 29 Descripción de otros

2.6. RESULTADOS

Se realizó el proceso de reingeniería de la plataforma Calisoft, creando una nueva interfaz gráfica mucho más intuitiva y atractiva para el usuario final, además se integraron nuevas funciones como notificaciones en tiempo real vía email y por medio de la misma plataforma, secciones de ayuda que informan de las múltiples funciones de cada vista, componentes que facilitan su uso y su visualización, además de integrar dos nuevos módulos externos que son; el módulo de

nomenclatura de bases de datos y el módulo de codificación, ambos hacen que el sistema sea integro a la hora de evaluar la calidad y usabilidad de un producto software. Las siguientes imágenes corresponden a la página principal y el dashboard de un usuario con rol de evaluador del aplicativo anterior y del aplicativo nuevo.



Imagen 9 Página principal del antiguo Calisoft



Imagen 10 Dashboard de evaluador del antiguo Calisoft



Imagen 11 Página principal del nuevo Calisoft

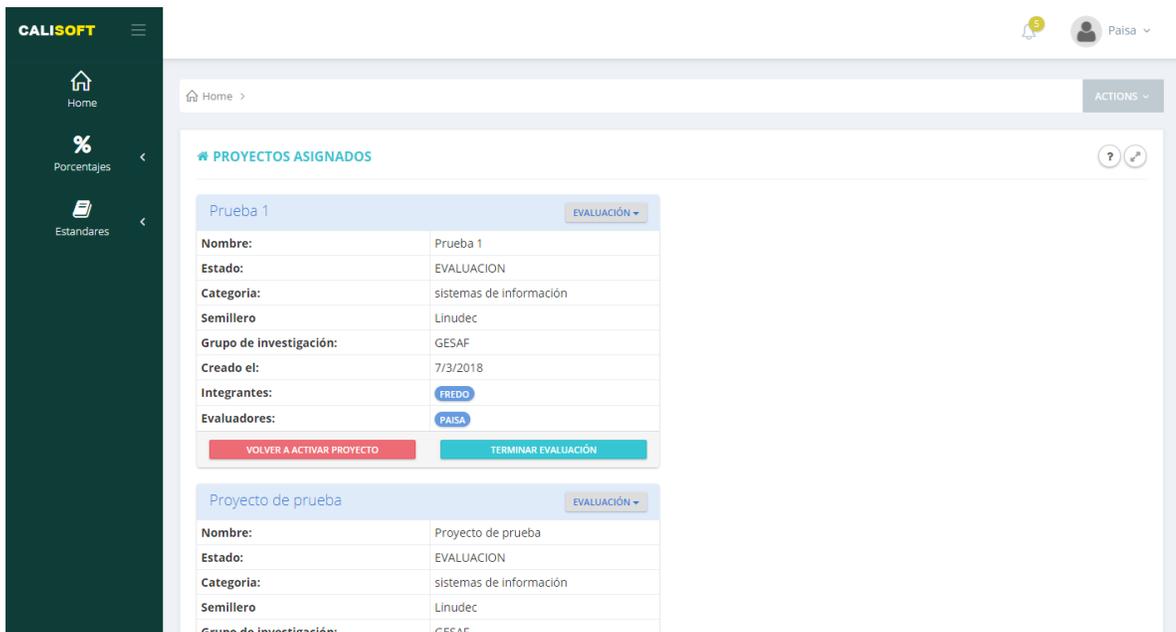


Imagen 12 Dashboard de evaluador del nuevo Calisoft

El nuevo Calisoft cuenta con los colores más opacos que genera un descanso en la vista del usuario final, además, cuenta con las opciones necesarias bien organizadas tanto en la página principal como en sus dashboards. Respecto al

módulo de Testing, la acción de crear casos de prueba es más eficiente, informativa y fácil de usar, como se puede ver a continuación;



Imagen 13 Vista casos de prueba, antiguo Calisoft

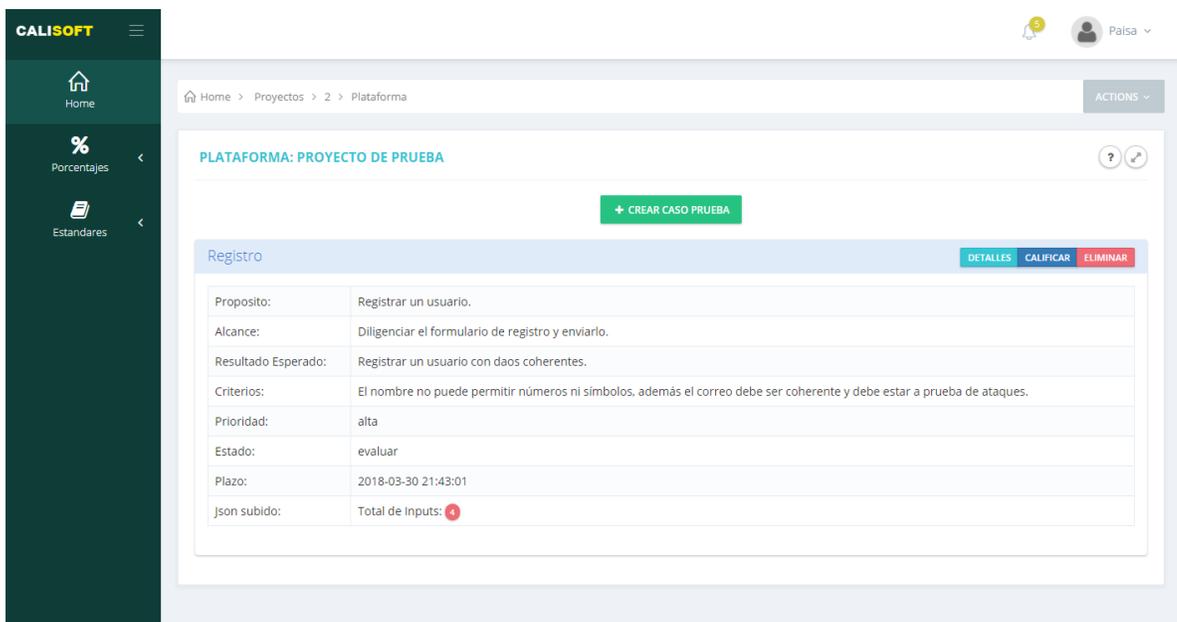


Imagen 14 Vista casos de prueba, nuevo Calisoft

La vista para crear casos de prueba en el nuevo Calisoft es más entendible, los casos de prueba de un proyecto se enlistan como bloques rectangulares y por medio del botón de "DETALLES" se pueden desplegar y volver a esconder los detalles de

un caso prueba sin la necesidad de dirigirse a otra vista como se hacía en el anterior Calisoft. Lo mismo sucede para crear un nuevo caso prueba;



Imagen 15 Crear caso prueba, antiguo Calisoft

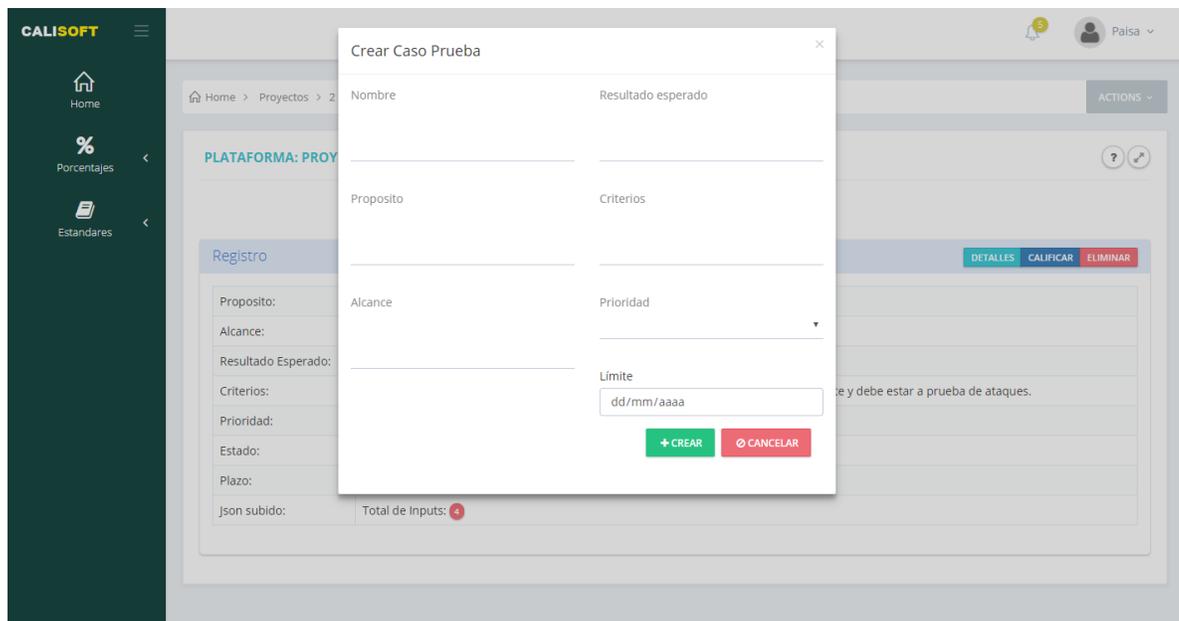


Imagen 16 Crear caso prueba, nuevo Calisoft

No es necesario dirigirse a una nueva vista como lo hacía el anterior Calisoft, ahora con el nuevo Calisoft se despliega un modal sobre la misma vista, eliminando la necesidad de ir a otra vista, y en el momento en que se crea, automáticamente se agrega al listado de casos de uso sin la necesidad de recargar la página.

En el caso para subir el formulario, no hace falta grabar una prueba y subir el archivo de la prueba, con el nuevo Calisoft basta con subir un texto JSON que lo da la misma extensión que se creó para este propósito.

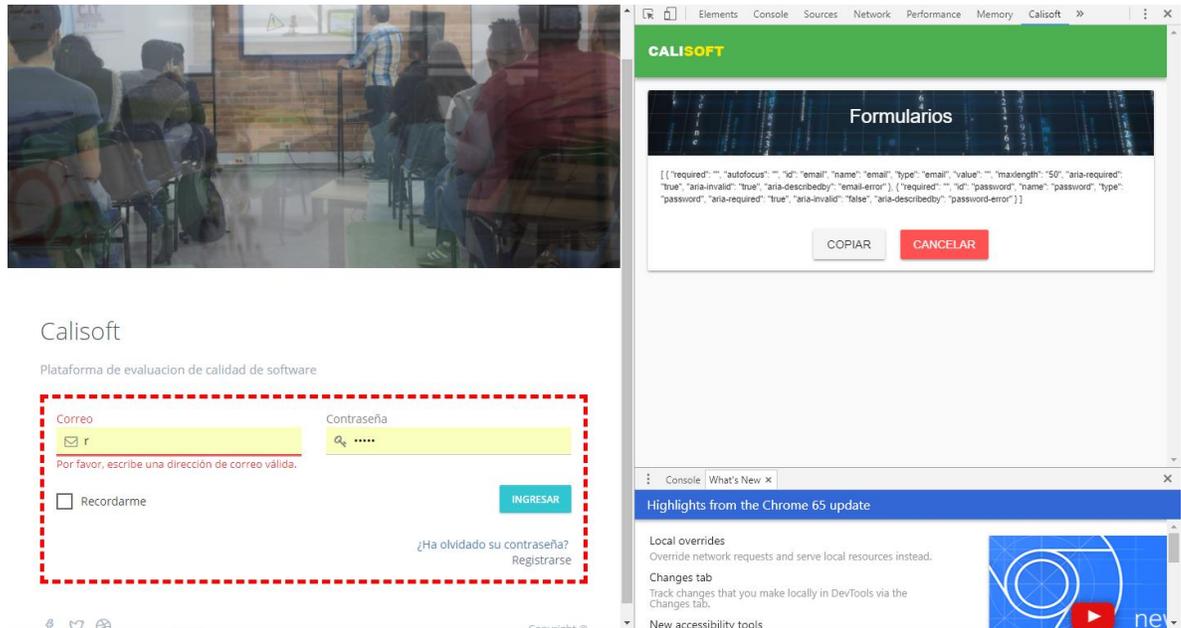


Imagen 17 Uso de la extensión, nuevo Calisoft

E ingresando únicamente el texto JSON en un input de tipo Texto, se sube el formulario al sistema para que el evaluador pueda hacer el procedimiento de evaluación.

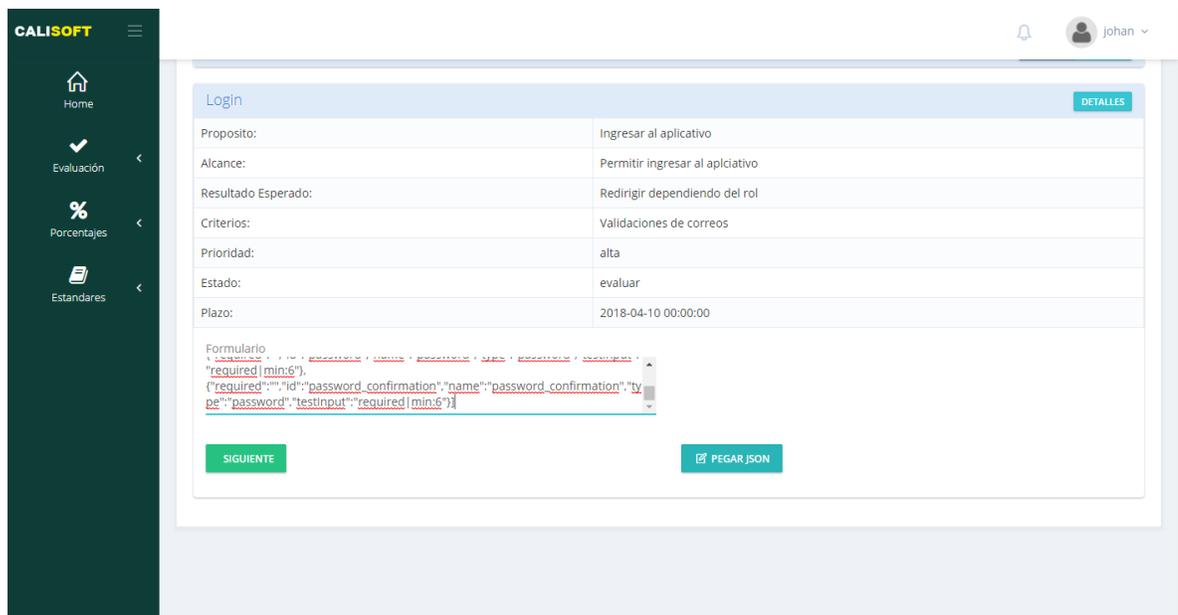


Imagen 18 Subir formulario, nuevo Calisoft



Imagen 19 Subir formulario, antiguo Calisoft

Cuando se sube el archivo, el evaluador procede a realizar la evaluación como se puede observar en las imágenes anteriores, no hay necesidad de subir un archivo para evaluar cualquier formulario de cualquier proyecto.

La vista de evaluación también fue optimizada si se compara con la anterior, la estructura es de la siguiente forma;

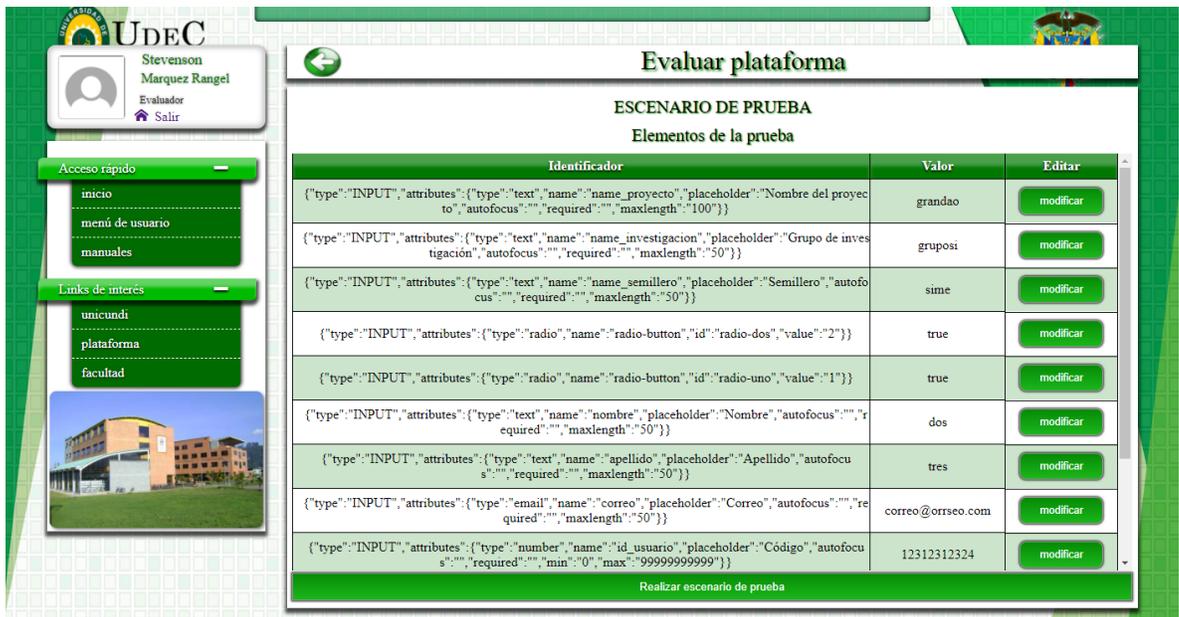


Imagen 20 Módulo de Testing, antiguo Calisoft

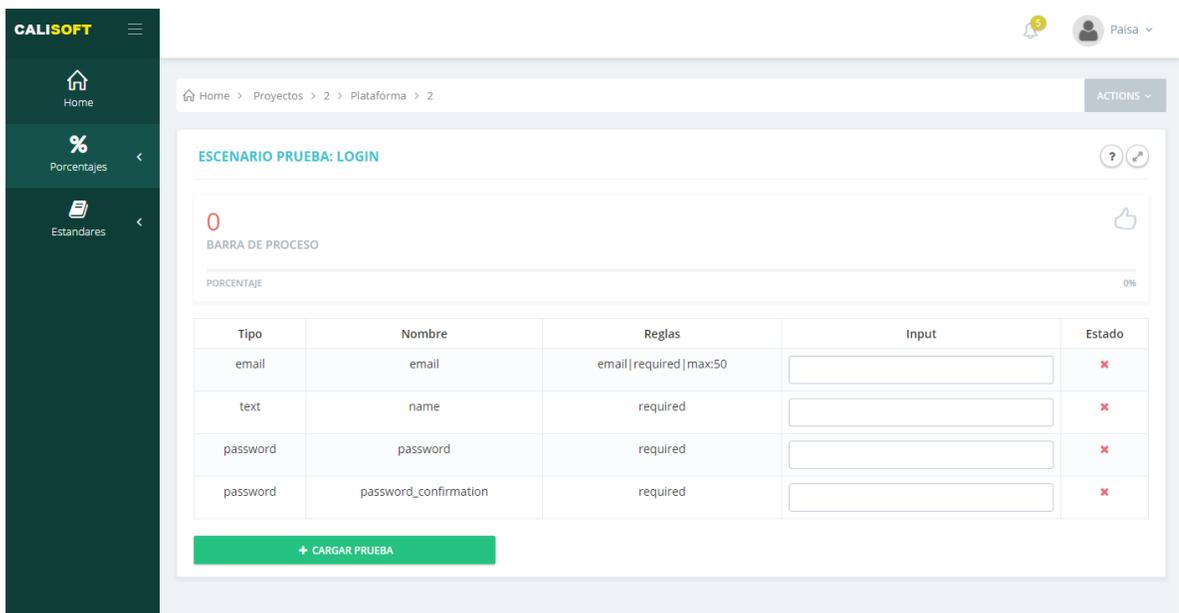


Imagen 21 Módulo de Testing, nuevo Calisoft

Se omiten los tipos de inputs innecesarios, como los checkbox, radio buttons y entre otros, con ello obtenemos una vista más clara de los inputs que verdaderamente nos interesan, que son los que permiten el ingreso de información errónea y maliciosa al sistema, en este paso no hay que hacer uso de la extensión para realizar la prueba, como sucedía en el antiguo Calisoft, sólo basta con oprimir el botón “CARGAR PRUEBA” y el sistema de forma automática ingresará valores

correctos e incorrectos y dará una calificación de manera inmediata, como se ve a continuación;

Tipo	Nombre	Reglas	Input	Estado
email	email	email required max:50	sborrego@latinmail.com	✓
text	name	required	dolore	✓
password	password	required	✓
password	password_confirmation	required	✓

CALIFICACIÓN 100%

Imagen 22 Procedimiento de Testing automatizado, nuevo Calisoft

El procedimiento es mucho más corto, se ahorran pasos y el proceso de evaluación se hace automático asegurando un estándar para los demás formularios que se evalúen por medio de Calisoft. Antes este proceso de evaluación podía estar tardando más de dos días, de esta nueva forma el proceso de evaluación dura un día dependiendo también de la participación del desarrollador.

2.7. CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- De acuerdo a los resultados se puede concluir que el tiempo en que se desarrollan las pruebas de Testing disminuyó más de la mitad, pasando de más de dos días a tan sólo un día, siempre y cuando haya una participación activa del desarrollador.
- La automatización de las pruebas permite que haya un estándar en el proceso de evaluación, ya no es necesario la creatividad o experiencia del evaluador para probar múltiples opciones datos correctos e incorrectos sobre cualquier formulario.

- La calidad de evaluación aumentó, dado que se disminuyó tiempos de evaluación, tiempos de respuesta y se optimizaron procesos que anteriormente eran un poco largos y complejos.
- El sistema hace uso de usabilidad, dado que su nueva estructura es amigable con el usuario final, informando cuales son los pasos que debe seguir. Además, el usuario puede encontrar múltiples funciones organizadas dentro de una misma vista, esto permite comodidad al no tener que desplazarse por diferentes vistas.
- La integración de módulos externos hizo que el proceso de evaluación fuera más riguroso para el desarrollador, dado que debe implementar los estándares estipulados en cada módulo. Esto ayudo bastante a aumentar la calidad de los productos software que evaluó el aplicativo, dado que todos manejan el mismo estándar certificando una buena estructura en su programación.
- Para la elaboración de este módulo, se implementó dos plugin que son base en su funcionamiento, el plugin Vee.Validate de Vue.js, encargado de validar en tiempo real un input con base a las reglas de validación implementadas por el desarrollador, y el plugin Laravel Faker, encargado de ingresar información proveniente de la base de datos de manera automática, este proceso se realiza sobre los sobre los inputs obtenidos del formulario que ofrece el desarrollador.

RECOMENDACIONES

- Se recomienda hacer uso del navegador Google Chrome, dado que la extensión únicamente trabaja sobre ese navegador, de tal forma que, si desea hacer uso de la extensión para obtener los atributos de cualquier formulario, debe uso de ese navegador.
- Se recomienda para una próxima actualización, la extensión pueda obtener las validaciones que no están expresadas en el código html, y de forma dinámica pueda acceder a la base de datos del desarrollador por medio de permisos y generar pruebas directamente al servidor. Así se podría evaluar temas como cantidades máximas, nombres repetidos, validaciones más personalizadas, entre otras.
- Se recomienda incorporar un chat que permita una comunicación más directa entre el desarrollador y el evaluador, dado que se dio un buen paso con el sistema de notificaciones, pero sí hace falta ese módulo, dado que, para solicitar una modificación o cierta acción se debe hacer por medio de correo o de forma presencial.
- Se recomienda para una próxima actualización, idear una lógica que permita realizar el sistema de Testing de manera secuencial y por tiempos, sin la necesidad de cargar la prueba y guardar el resultado de forma manual.

2.8. BIBLIOGRAFÍA

- Bolaños-Pizarro, M., & Vidal-Infer, A; Navarro-Molina, Carolina; Valderrama-Zuriañ, Juan Carlos; Aleixandre-Benavent, R. (2007). Usabilidad: concepto y aplicaciones en las páginas web médicas. *Papeles Médicos*, Vol 16, Nr(1), 14–21. Retrieved from http://sedom.es/wp-content/themes/sedom/pdf/4cbd571ba0446art_3.pdf
- Carmona, J. F. S. (2016). Diseño e implementación de un sistema de testing automatizado y con validación de web components integrados en aplicaciones web. *Universidad Politécnica de Madrid Escuela Técnica Superior de Ingenieros Informáticos*, 112.
- Centro Nacional de Información de Ciencias Médicas, L., & Moráquez Bergues, M. (2013). *Revista cubana de información en ciencias de la salud. Revista Cubana de Información en Ciencias de la Salud* (Vol. 24). 2002, Editorial Ciencias Médicas. Retrieved from http://scielo.sld.cu/scielo.php?pid=S2307-21132013000200007&script=sci_arttext&tlng=en
- Expo, Q. (2014). Conferencia internacional de Testing y calidad de Software. Retrieved March 5, 2017, from <http://www.expoqa.com/pdf/expoqa14/ExpoQA-Programa-14-ES-V02.pdf>
- Mascheroni, M. A., Greiner, C. L., Dapozo, G. N., & Estayno, M. G. (2014). Ingeniería de Usabilidad. Una Propuesta Tecnológica para Contribuir a la Evaluación de la Usabilidad del Software. *Revista Latinoamericana de Ingeniería de Software*, 1(4), 125. <https://doi.org/10.18294/relais.2013.125-134>
- Mascheroni, M., Greiner, C., & Petris, R. (2012). Calidad de software e Ingeniería de Usabilidad. In *WICC 2012* (pp. 656–659).
- Sánchez, W. O., & Ovidio, W. (2013). La usabilidad en Ingeniería de Software: definición y características. Retrieved from <http://www.redicces.org.sv/jspui/handle/10972/1937>
- 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>
- Ferré, X., & Sánchez, M. (2002). Desarrollo Orientado a Objetos con UML. *Facultad de Informática- UPM*, 2, 1–38. Retrieved from <http://rafaelmellado.cl/material/com3162/complementario/05.pdf>
- FOWLER, M. (1999). *UML gota a gota*. México: Addison Wesley Longman de

México. S.A. de C.V.

García, J., Ortín, M. J., Moros, B., Nicolás, J., & Tova, A. (2007). De los Procesos del Negocio a los Casos de Uso. *Cyta*, 6(4). Retrieved from [http://amorena.com.ar/PAG DE MATERIAS Y LIBROS/LIBROS TODOS/MODELADO DE NEGOCIO EN CASO DE USO.pdf](http://amorena.com.ar/PAG_DE_MATERIAS_Y_LIBROS/LIBROS_TODOS/MODELADO_DE_NEGOCIO_EN_CASO_DE_USO.pdf)

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

Marcial, C., Carmona, R., & Milenio, R. T. (n.d.). Matemáticas I. Retrieved from https://s3.amazonaws.com/academia.edu.documents/38508572/Matematicas_I-Parte1.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1523160600&Signature=WHNdDUS7%2F7%2FphArrF5OPmZNiSVc%3D&response-content-disposition=inline%3B filename%3DMatematicas_I-Parte1.pdf

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

Sanchis, R., Poler, R., & Ortiz, Á. (2009). Técnicas para el Modelado de Procesos de Negocio en Cadenas de Suministro TT - Business Process Modelling Techniques in Supply Chains. *Información Tecnológica*, 20(2), 29–40. <https://doi.org/10.4067/S0718-07642009000200005>

2.9. ANEXOS

2.9.1.MANUAL DE INSTALACIÓN

MANUAL DE INSTALACIÓN

CONTENIDO

INTRODUCCIÓN	68
REQUERIMIENTOS MÍNIMOS DEL SISTEMA	68
PROGRAMAS NECESARIOS	68
COMPOSER	68
XAMMP 70	
NODE.JS 73	
GOOGLE CHROME.....	74
INSTALACIÓN	75
Uso de XAMPP para ubicar la carpeta del proyecto	75
Instalación de Composer en la carpeta del proyecto	77
Creación de la base de datos	81
Modificación del archivo example.env	82
Configuración del repositorio de MAIL.....	84
Configuración de Pusher para notificaciones en tiempo real	88
Cambiar nombre archivo exmaple.env	93
Llave de acceso	95
Creación ruta a carpeta Storage	95
Instalación de Vue de Node.js en la carpeta del proyecto	96
Migraciones.....	97
Seeders 99	
Correr el aplicativo.....	100
Acceso al login.....	100
REFERENCIAS.....	102

ILUSTRACIONES

Ilustración 1	Página principal de Composer	68
Ilustración 2	Documentación de instalación de Composer	69
Ilustración 3	Instalación de Composer	70
Ilustración 4	Página principal de XAMPP	71
Ilustración 5	Recomendaciones antes de la instalación de XAMPP	72
Ilustración 6	Proceso de la instalación de XAMPP	72
Ilustración 7	Página principal de node.js	73
Ilustración 8	Proceso de instalación de Node.js	74
Ilustración 9	Página de descarga de Google Chrome	75
Ilustración 10	Ruta de la carpeta principal de Calisoft	75
Ilustración 11	Interfaz XAMPP.....	76
Ilustración 12	Interfaz XAMPP con servicios activados	77
Ilustración 13	Ventana ejecutar.....	77
Ilustración 14	Simbolo del sistema	78
Ilustración 15	Ejecución del comando "cd.."	78
Ilustración 16	Ejecución del comando "cd.."	78
Ilustración 17	Cmd ubicación de la carpeta principal.....	79
Ilustración 18	Composer install en CMD	79
Ilustración 19	Procedimiento de la isntalación de Composer en carpeta principal.....	79
Ilustración 20	Finalización de Composer en carpeta principal.....	80
Ilustración 21	Comando Composer update en CMD	80
Ilustración 22	Actualización Composer en CMD.....	81
Ilustración 23	Creación base de datos.....	82
Ilustración 24	Ubicación archivo "excmple.env".....	82
Ilustración 25	Contenido archivo "example.env"	83
Ilustración 26	Configuración de MAIL.....	84

Ilustración 27	Página principal de Mailtrap.....	84
Ilustración 28	Inició de sesión Mailtrap.....	85
Ilustración 29	Backend Mailtrap	85
Ilustración 30	Administración de mensajes de Mailtrap	86
Ilustración 31	Apartado Integrations Mailtrap.....	87
Ilustración 32	Atributos de Mailtrap en Laravel.....	88
Ilustración 33	Apartado de Mail en archivo example.env	88
Ilustración 34	Página principal de Pusher	89
Ilustración 35	Welcome to Pusher.....	90
Ilustración 36	Modificación de Pusher	91
Ilustración 37	Backend Pusher	92
Ilustración 38	Pestañas Pusher	92
Ilustración 39	App Keys Pusher	93
Ilustración 40	Configuración Pusher en archivo example.env	93
Ilustración 41	Cambiar nombre del archivo example.env	94
Ilustración 42	Archivo. env	95
Ilustración 43	Php artisan key:generate	95
Ilustración 44	Proceso exitoso del comando Php artisan storage:link	95
Ilustración 45	Proceso de instalación de Vue	96
Ilustración 46	Finalización exitosa de Vue en carpeta del proyecto	97
Ilustración 47	Comando php artisan migrate.....	98
Ilustración 48	Migraciones exitosas.....	98
Ilustración 49	Comando php artisan db:seed exitoso	99
Ilustración 50	Registros de la tabla TBL_Usuarios	99
Ilustración 51	Datos de los seeders	100
Ilustración 52	Ip del servidor local	101
Ilustración 53	Página principal de Calisoft.....	101

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.

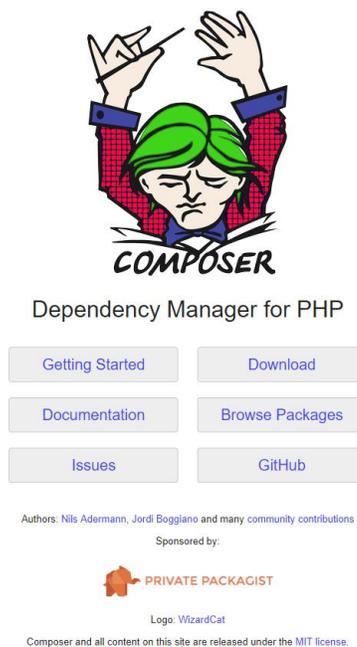


Ilustración 1 Página principal de Composer

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

[Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

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;

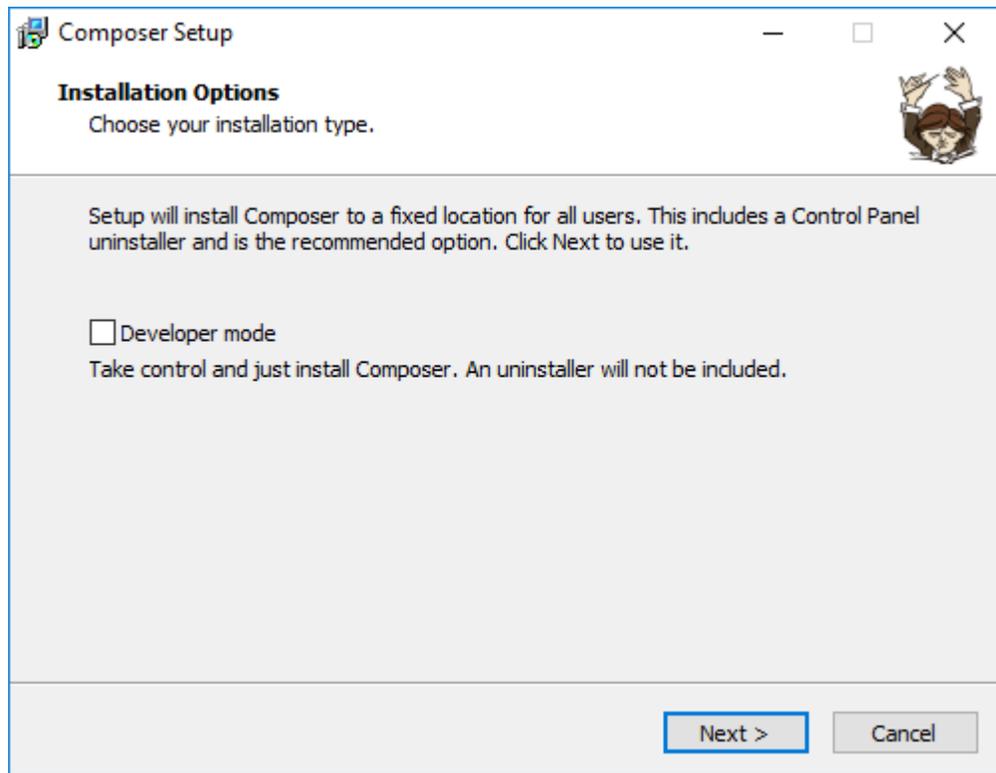


Ilustración 3 Instalación de Composer

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.

XAMMP

“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;

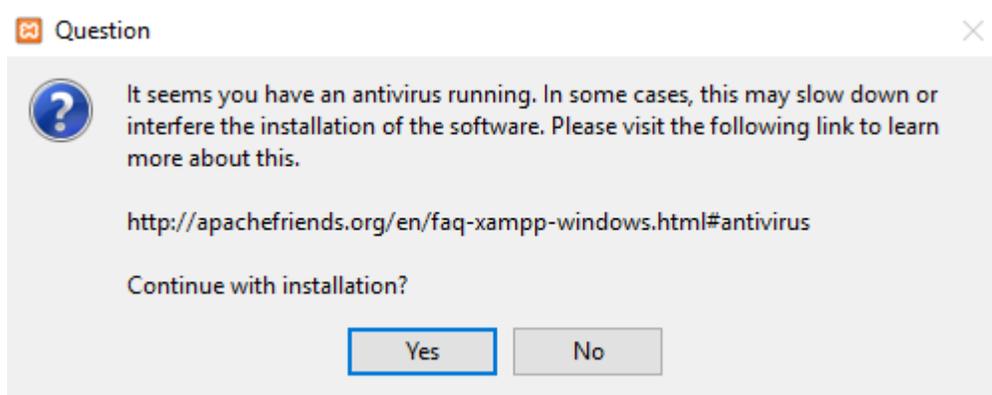


Ilustración 5 Recomendaciones antes de la instalación de XAMPP

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;

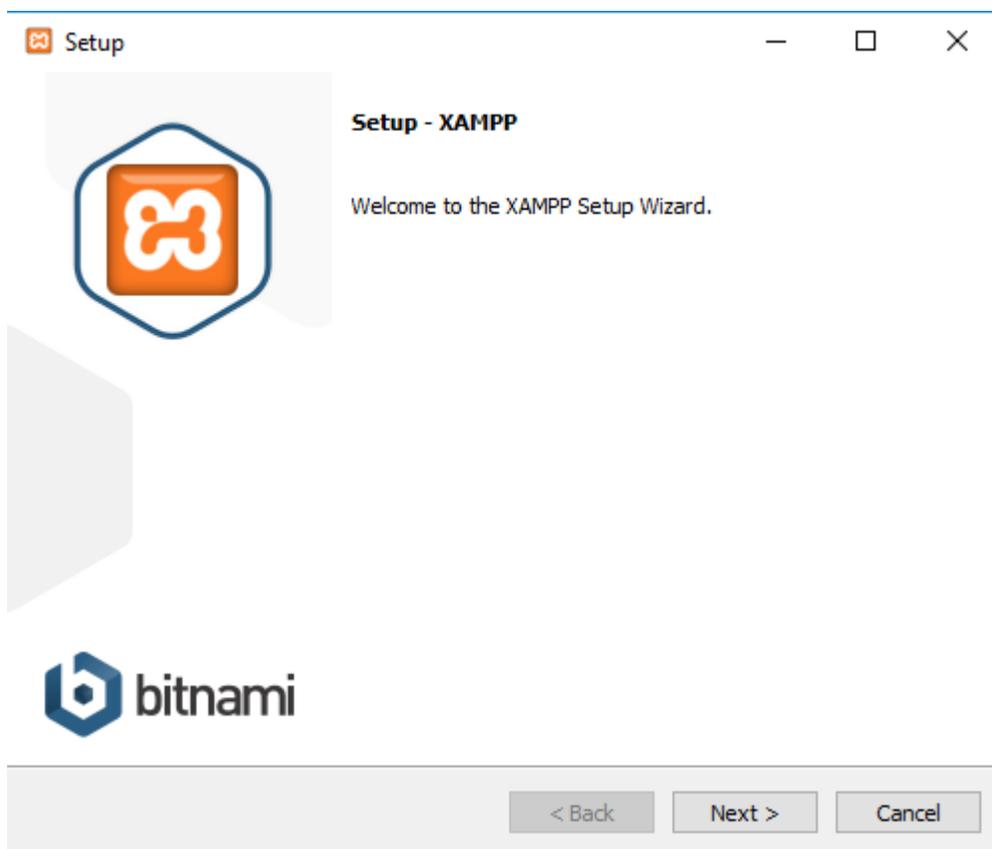


Ilustración 6 Proceso de la instalación de XAMPP

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)



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

Ó revise la [Agenda de LTS](#).

1. *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;

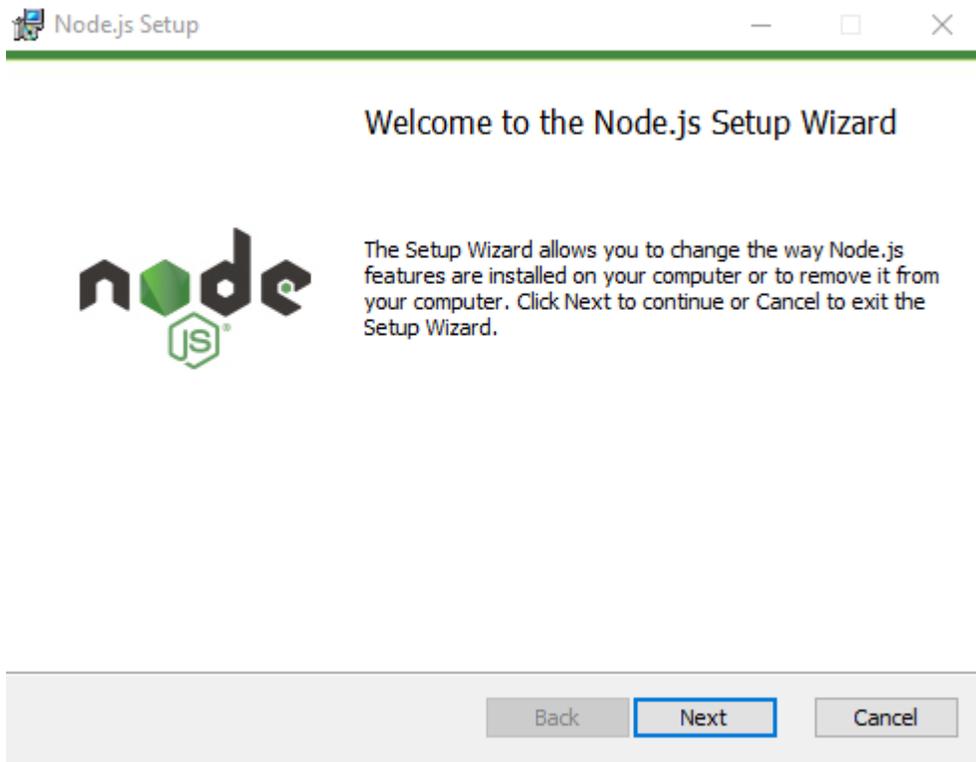


Ilustración 8 Proceso de instalación de Node.js

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

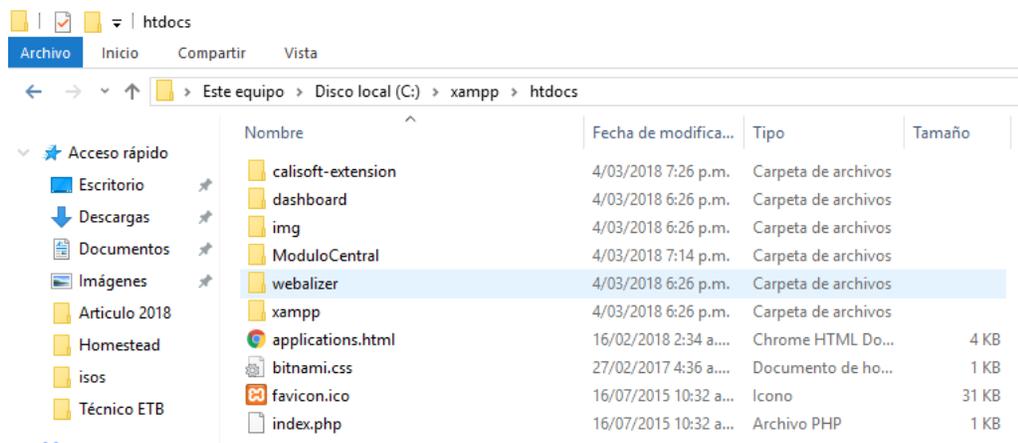


Ilustración 10 Ruta de la carpeta principal de Calisoft

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;

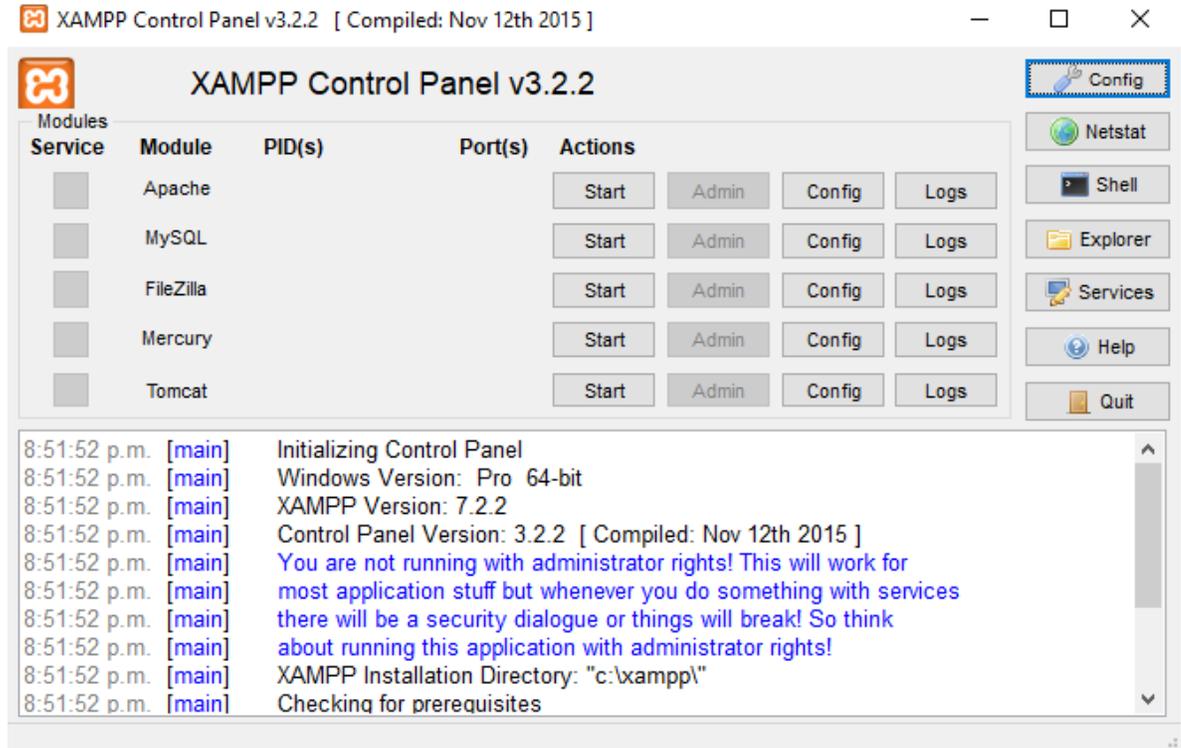


Ilustración 11 Interfaz XAMPP

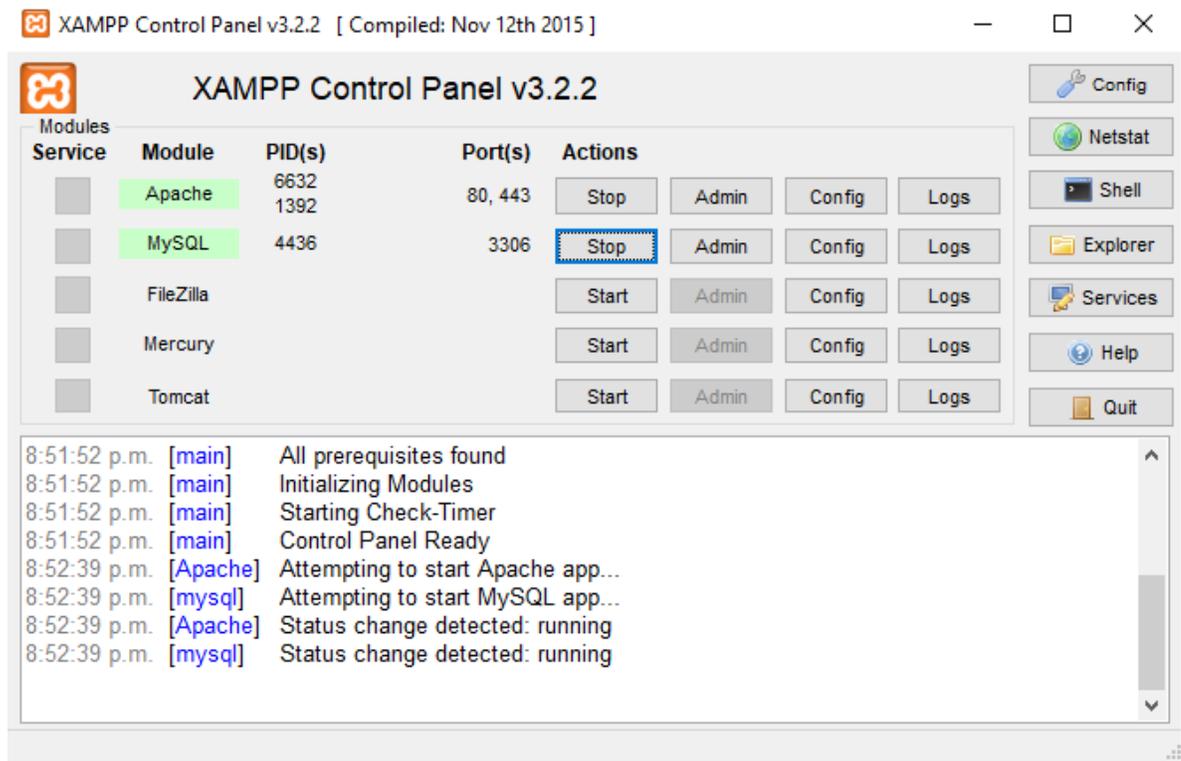


Ilustración 12 Interfaz XAMPP con servicios activados

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"

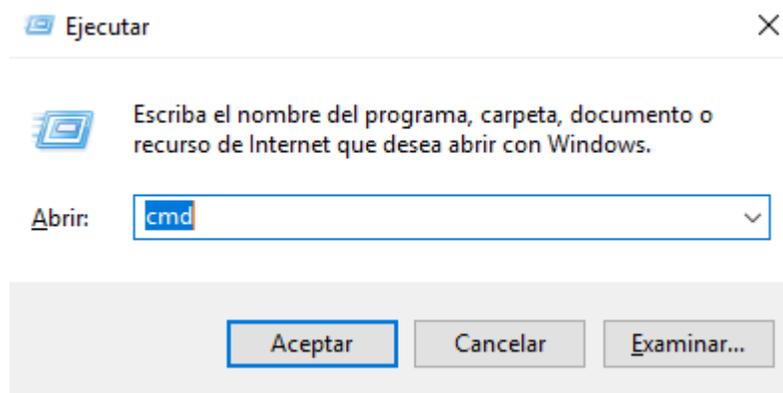
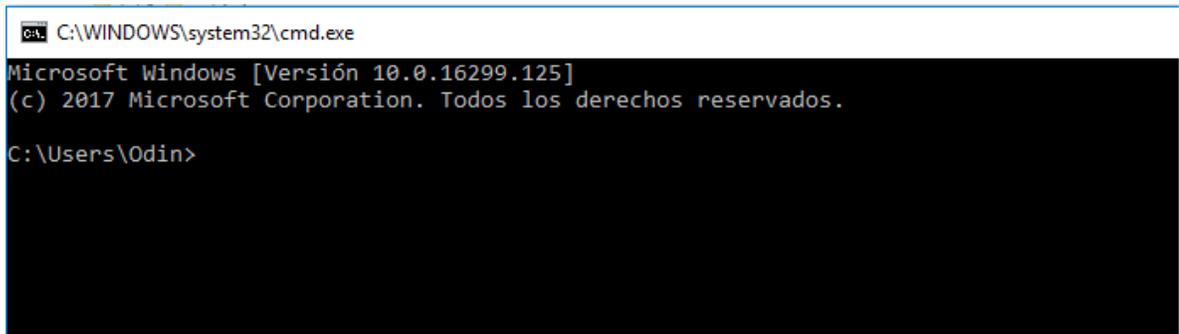


Ilustración 13 Ventana ejecutar

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

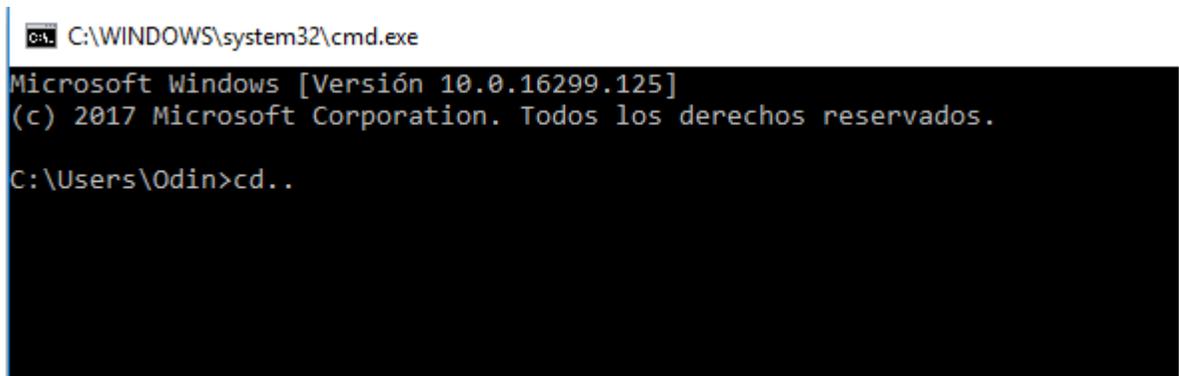


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.16299.125]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Odin>
```

Ilustración 14 Símbolo del sistema

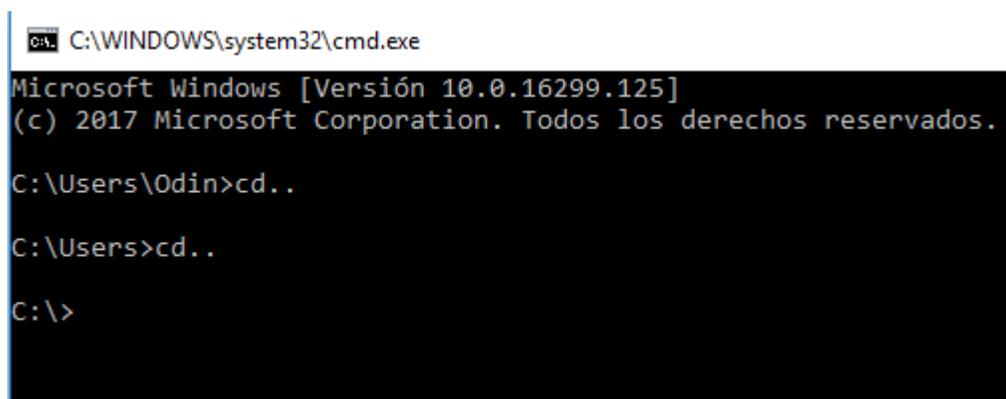
Se nos abrirá el símbolo del sistema ubicado en la carpeta del usuario del ordenador, en este caso el usuario principal es “Odin”, por medio del comando “cd..” retrocedemos al disco local “C”.



```
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
symfony/http-kernel suggests installing symfony/browser-kit ()
symfony/http-kernel suggests installing symfony/class-loader ()
symfony/http-kernel suggests installing symfony/config ()
symfony/http-kernel suggests installing symfony/dependency-injection ()
symfony/console suggests installing symfony/filesystem ()
egulias/email-validator suggests installing ext-intl (PHP Internationalization Libraries are required to use the SpoofChecking validation)
paragonie/random_compat suggests installing ext-libsodium (Provides a modern crypto API that can be used to generate random bytes.)
ramsey/uuid suggests installing ext-libsodium (Provides the PECL libsodium extension for use with the SodiumRandomGenerator)
ramsey/uuid suggests installing ext-uuid (Provides the PECL UUID extension for use with the PeclUuidTimeGenerator and PeclUuidRandomGenerator)
ramsey/uuid suggests installing ircmaxell/random-lib (Provides RandomLib for use with the RandomLibAdapter)
ramsey/uuid suggests installing moonbloat/math (Provides support for converting UUID to 128-bit integer (in string form).)
ramsey/uuid suggests installing ramsey/uuid-console (A console application for generating UUIDs with ramsey/uuid)
ramsey/uuid suggests installing ramsey/uuid-doctrine (Allows the use of RamseyUuid\Uuid as Doctrine field type.)
symfony/translation suggests installing symfony/config ()
monolog/monolog suggests installing aws/aws-sdk-php (Allow sending log messages to AWS services like DynamoDB)
monolog/monolog suggests installing doctrine/couchdb (Allow sending log messages to a CouchDB server)
monolog/monolog suggests installing ext-amqp (Allow sending log messages to an AMQP server (1.0+ required))
monolog/monolog suggests installing ext-mongo (Allow sending log messages to a MongoDB server)
monolog/monolog suggests installing mongodb/mongodb (Allow sending log messages to a MongoDB server via PHP Driver)
monolog/monolog suggests installing php-amqplib/php-amqplib (Allow sending log messages to an AMQP server using php-amqplib)
monolog/monolog suggests installing php-console/php-console (Allow sending log messages to Google Chrome)
monolog/monolog suggests installing rollbar/rollbar (Allow sending log messages to Rollbar)
monolog/monolog suggests installing ruflin/elastica (Allow sending log messages to an Elastic Search server)
monolog/monolog suggests installing sentry/sentry (Allow sending log messages to a Sentry server)
league/flysystem suggests installing league/flysystem-aws-s3-v2 (Allows you to use S3 storage with AWS SDK v2)
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 server storage via phpseclib)
league/flysystem suggests installing league/flysystem-ziparchive (Allows you to use ZipArchive adapter)
league/flysystem suggests installing spatie/flysystem-dropbox (Allows you to use Dropbox storage)
league/flysystem suggests installing srmlive/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 use 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.)
preidis/preidis 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>
```

Ilustración 20 Finalización de Composer en carpeta principal

De esta forma se debe finalizar la instalación de Composer. A continuación, procedemos a ejecutar el comando “Composer update”, para que actualice su contenido.

```
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
```

Ilustración 21 Comando Composer update en CMD

Pulsamos la tecla enter y la actualización se generará automáticamente y deberá finalizar de la siguiente manera;

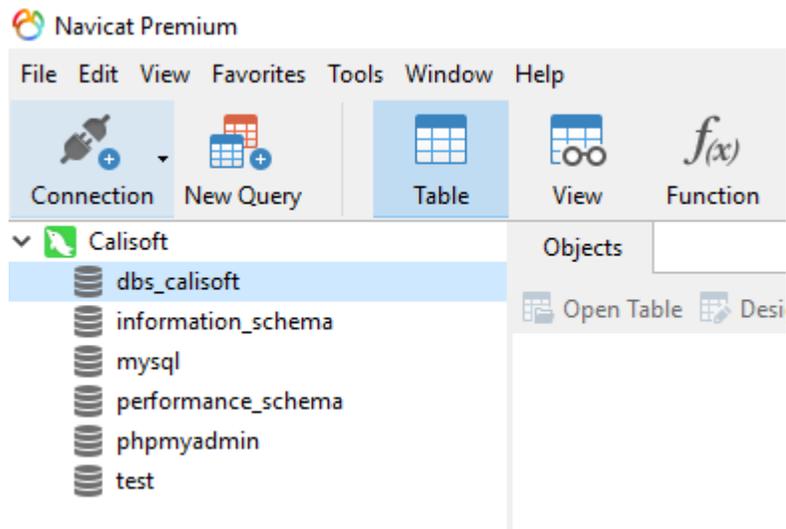


Ilustración 23 Creación base de datos

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`”.

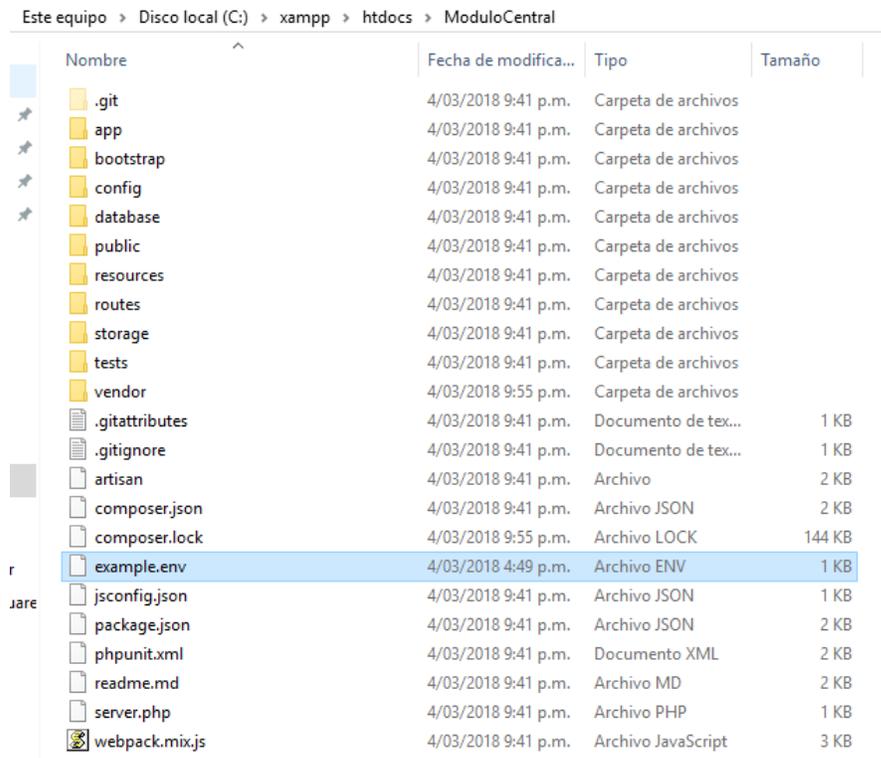


Ilustración 24 Ubicación archivo "example.env"

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.

3. 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.)

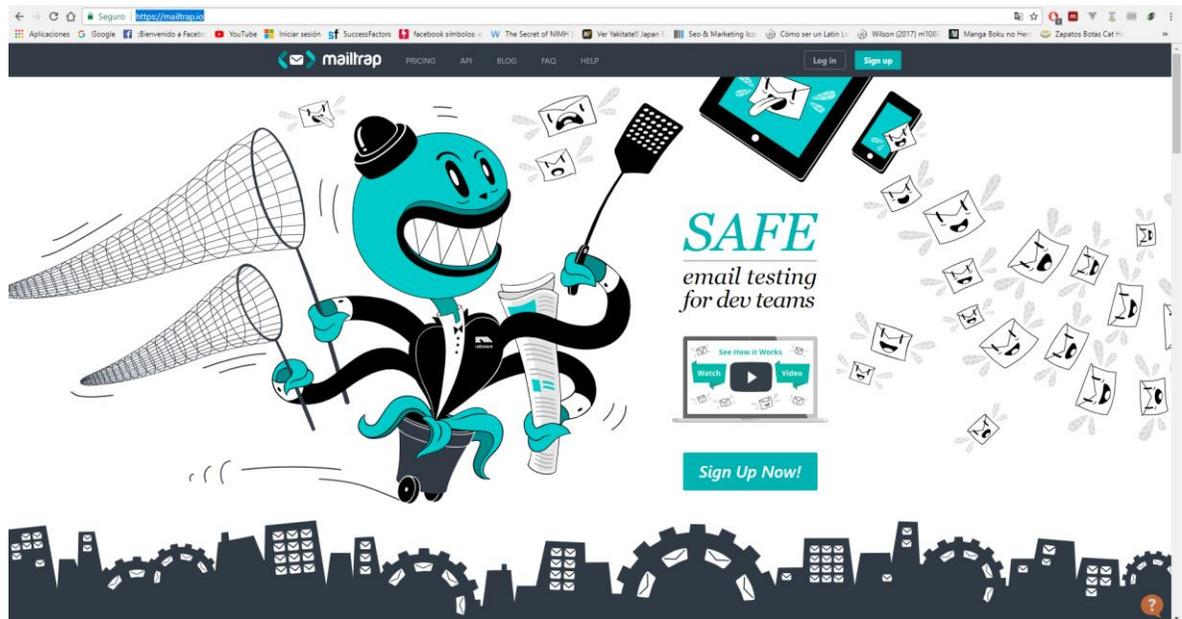


Ilustración 27 Página principal de Mailtrap

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

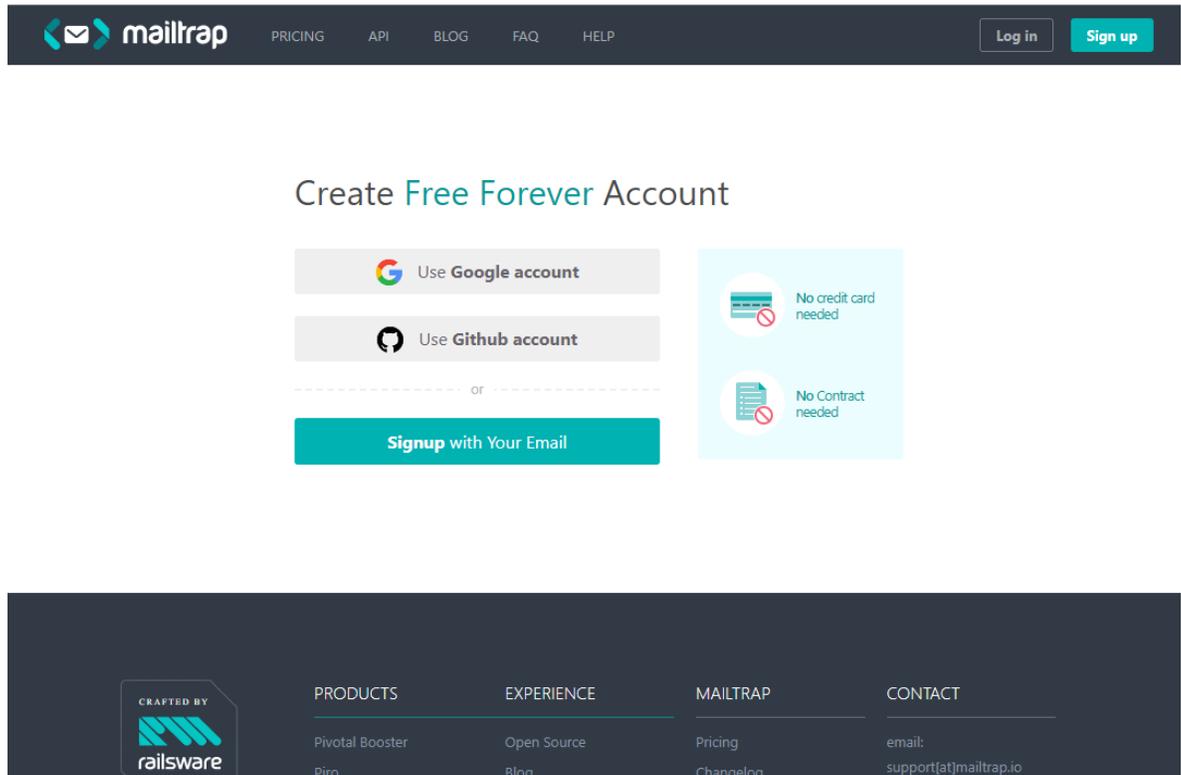


Ilustración 28 Inicio de sesión Mailtrap

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;

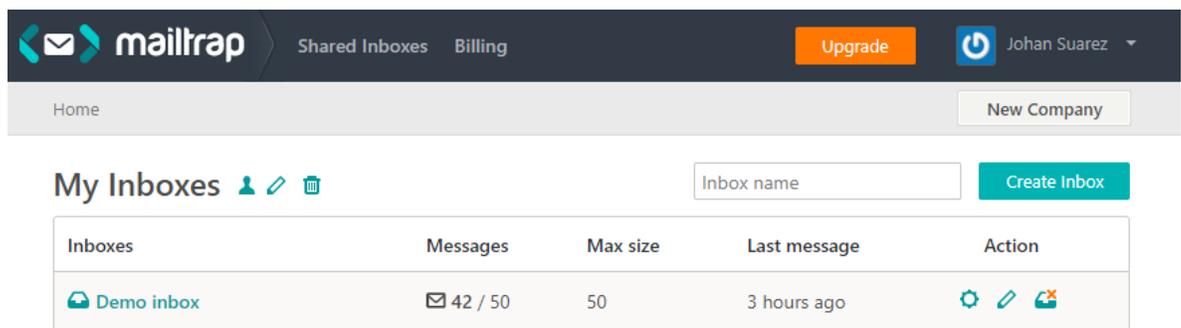


Ilustración 29 Backend Mailtrap

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, there is a list of messages with the following subjects and recipients:

- Se te ha asignado un evaluador (To: <fredo@joya.joya>)
- Se te ha asignado un proyecto (To: <paisa@mail.com>)
- Su proyecto fue aceptado ¡Felicitaciones! (To: <fredo@joya.joya>)
- Se ha enviado un formulario de un caso prueba (To: <paisa@mail.com>)
- Se ha creado un caso prueba para su proyecto (To: <fredo@joya.joya>)
- Proyecto listo para evaluación (To: <paisa@mail.com>)
- Se ha activado su proyecto (To: <fredo@joya.joya>)
- Se ha enviado un formulario de un caso prueba (To: <paisa@mail.com>)
- Se ha creado un caso prueba para su proyecto (To: <fredo@joya.joya>)
- Se ha enviado un formulario de un caso prueba (To: <paisa@mail.com>)
- Se ha creado un caso prueba para su proyecto (To: <fredo@joya.joya>)
- Se ha enviado un formulario de un caso prueba (To: <paisa@mail.com>)
- Se ha creado un caso prueba para su proyecto (To: <fredo@joya.joya>)
- Se ha enviado un formulario de un caso prueba (To: <paisa@mail.com>)

On the right, the 'SMTP Settings' section is active, showing the following configuration:

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

The 'Integrations' section has 'Ruby on Rails' selected in the dropdown menu. Below it, there is a code block for configuring ActionMailer defaults:

```

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

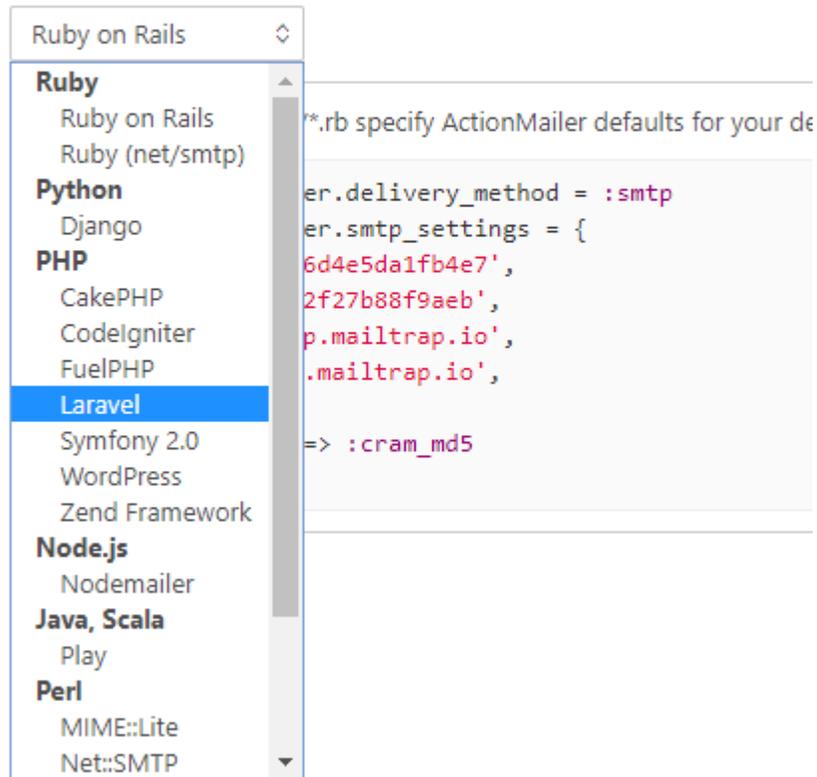


Ilustración 31 Apartado Integrations Mailtrap

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/>

New!  Chatkit, developer-driven chat done simply. [Learn more >](#)

 **PUSHER** [We're hiring](#)

[Features](#) [Use Cases](#) [Developers](#) [Pricing](#) [Blog](#) [Login](#) [Sign Up](#)

Build products with **realtime** features

We spend our time maintaining reliable and scalable realtime infrastructure so you can spend yours building awesome realtime features.

[Get in touch >](#)

Sign up with ...

 GitHub  Google

Or sign up with email

Email address

Password

No soy un robot  reCAPTCHA
Privacidad - Condiciones

[Create a FREE account](#)

You agree to the [Terms of Service](#) and [Privacy Policy](#).

   INTERCOM    & 200k Developers

Since 2011, they've been using our Data Delivery Network to get their app data to their Web, Mobile, IOT devices

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;

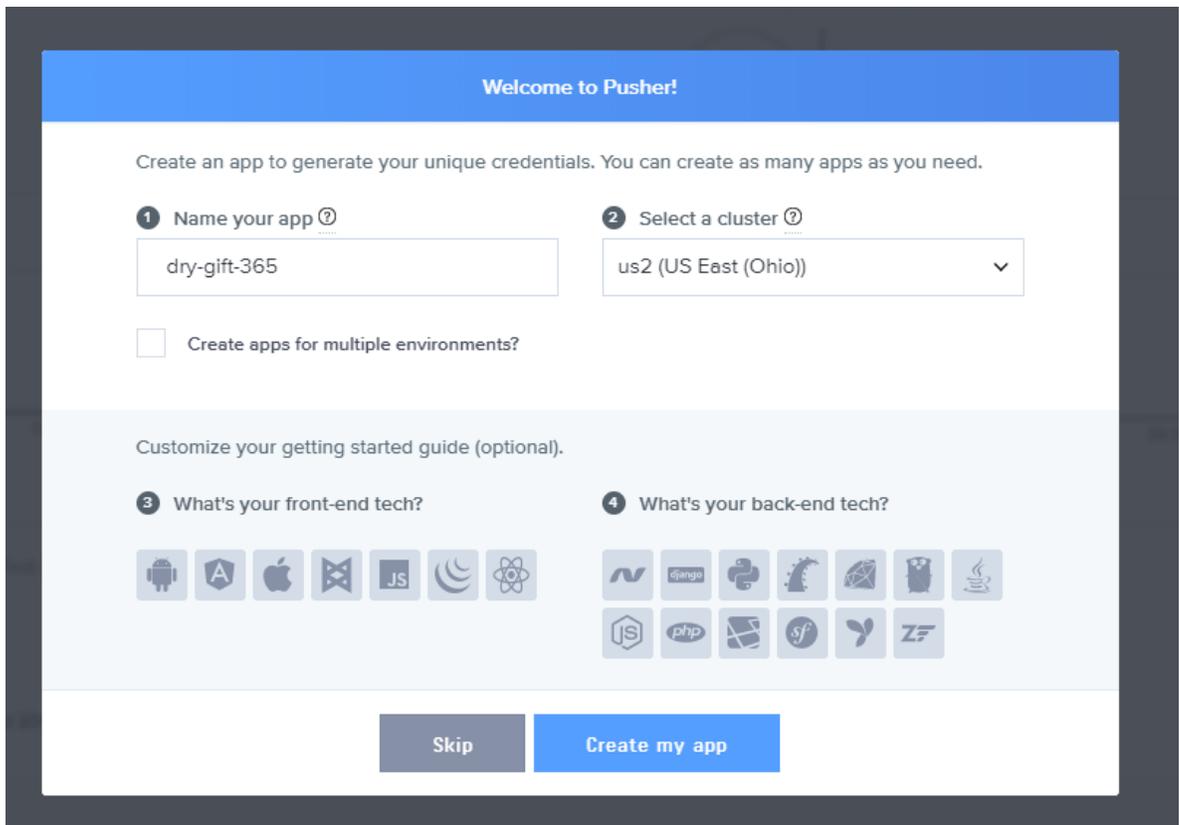


Ilustración 35 Welcome to Pusher

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.

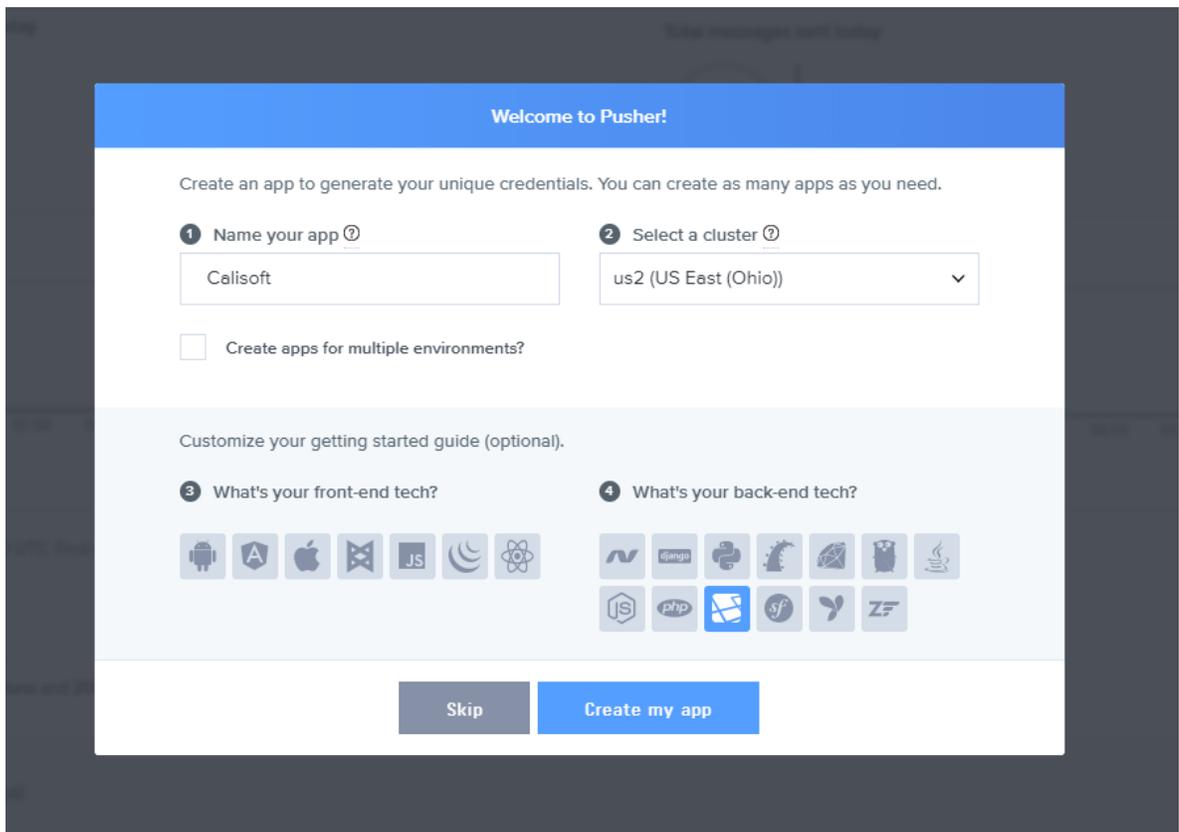


Ilustración 36 Modificación de Pusher

Para continuar se debe oprimir el botón “Create my app” y la página redirigirá a un Dashboard 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”

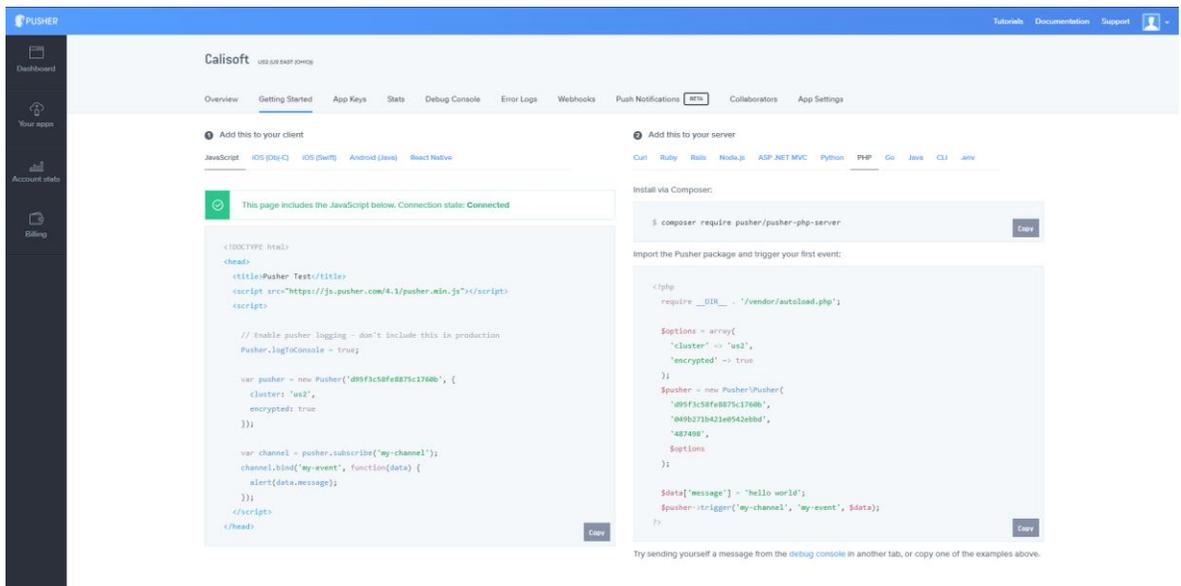


Ilustración 37 Backend Pusher

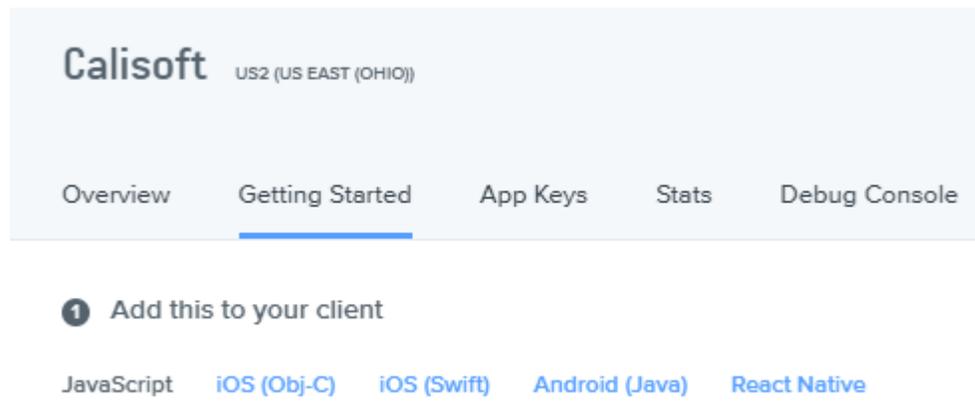


Ilustración 38 Pestañas Pusher

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

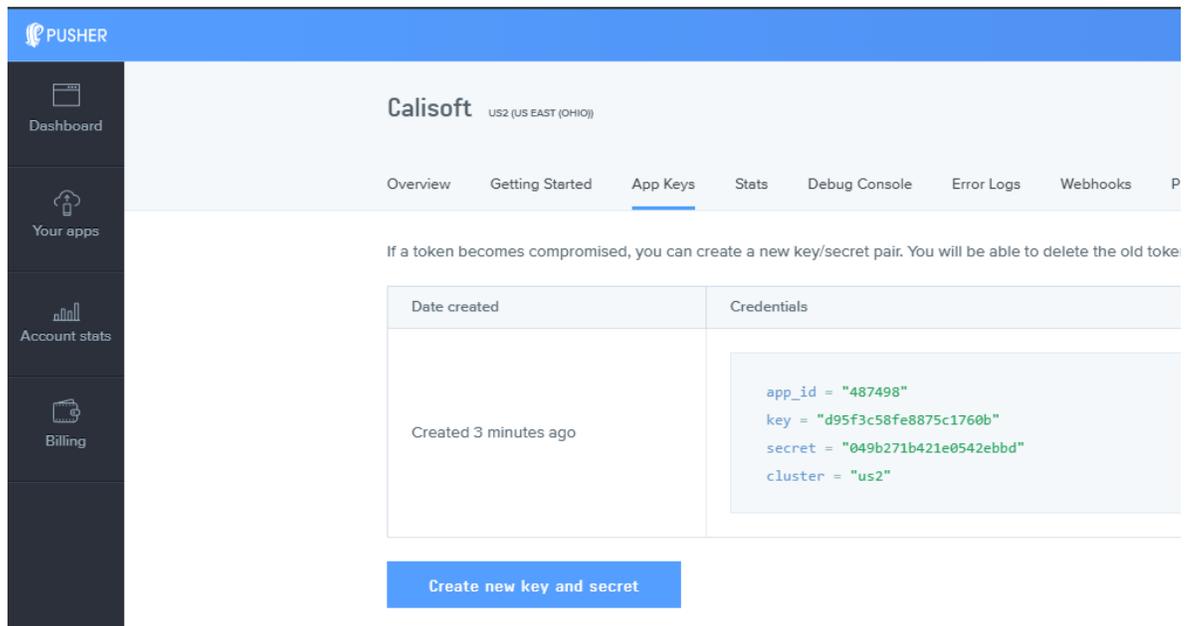


Ilustración 39 App Keys Pusher

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.

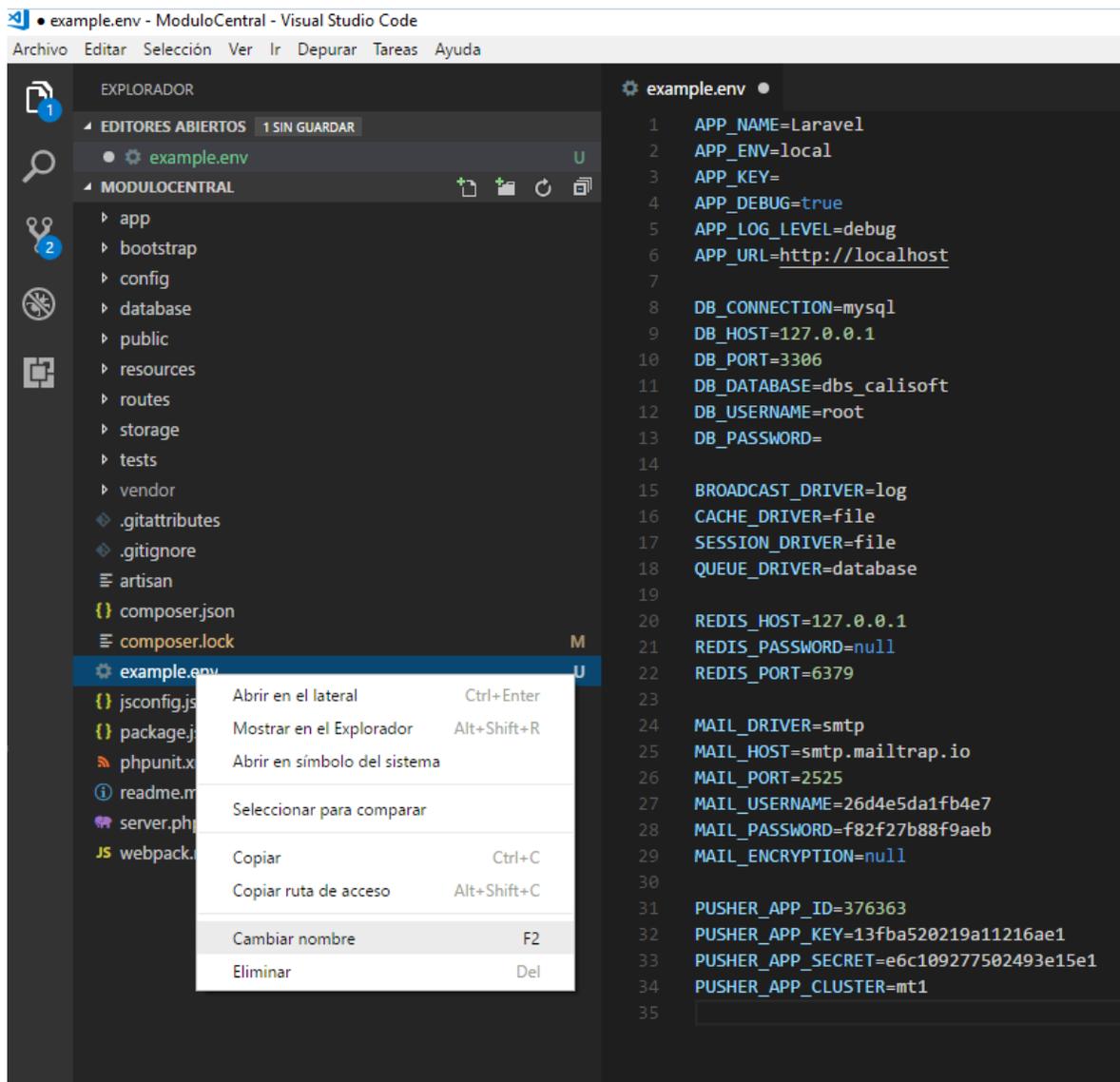


Ilustración 41 Cambiar nombre del archivo example.env

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

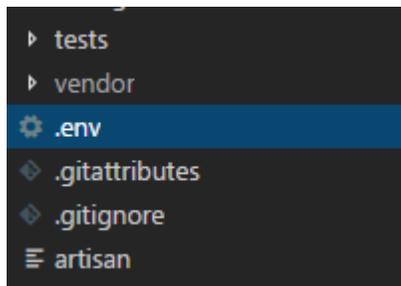


Ilustración 42 Archivo. env

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] [big] \js\tipo-nomenclatura
\js\nomenclaturas-show.js 253 kB   19 [emitted] [big] \js\nomenclaturas-show
\js\items-codificacion.js 192 kB   20 [emitted] [big] \js\items-codificacion
\js\categorias-show.js 254 kB   21 [emitted] [big] \js\categorias-show
\js\tipo-documento.js 192 kB   22 [emitted] [big] \js\tipo-documento
\js\semilleros.js 193 kB   23 [emitted] [big] \js\semilleros
\js\items-show.js 190 kB   24 [emitted] [big] \js\items-show
\js\notificaciones.js 189 kB   25 [emitted] [big] \js\notificaciones
\js\invitaciones.js 189 kB   26 [emitted] [big] \js\invitaciones
\js\bootstrap.js 189 kB   27 [emitted] [big] \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, diríjase a la base de datos y compruebe si las tablas, por ejemplo, TBL_Usuarios contienen registros.

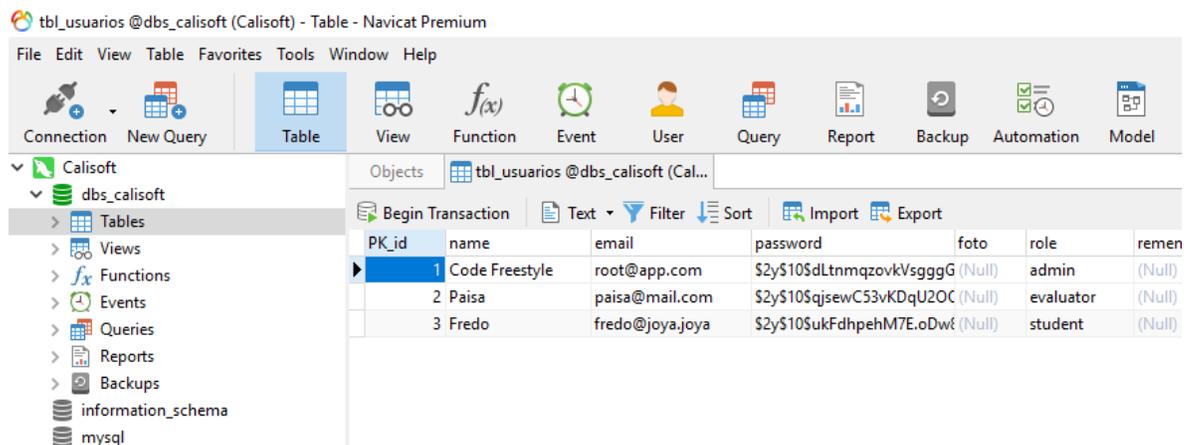


Ilustración 50 Registros de la tabla TBL_Usuarios

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

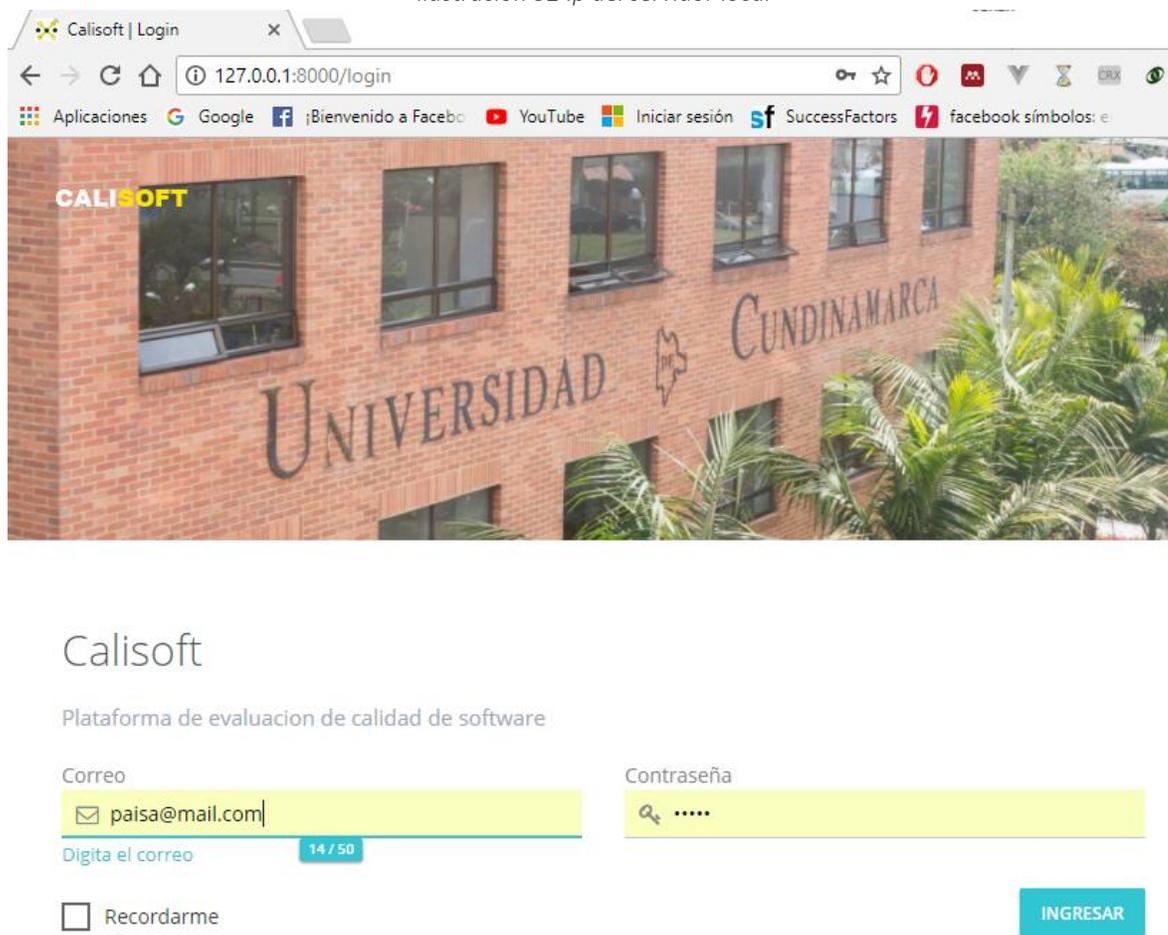


Ilustración 53 Página principal de Calisoft

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/>



3.1.1.MANUAL DE USUARIO

UDECA
UNIVERSIDAD DE
CUNDINAMARCA

CALISOFT UDECA



**DESARROLLO DE SOFTWARE
UNIVERSIDAD DE CUNDINAMARCA
MANUAL DE USUARIO**

2018

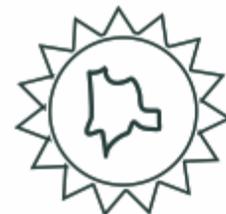


TABLA DE CONTENIDO

INTRODUCCIÓN.....	112
1. USUARIOS.....	113
1.1. ADMINISTRADOR	113
1.2. EVALUADOR	113
1.3. ESTUDIANTE	114
2. MÓDULOS	115
2.1. MODULOS BÁSICOS	¡Error! Marcador no definido.
2.2. MÓDULO DE MODELADO	115
2.3. MÓDULO DE TESTING	115
2.4. MÓDULO DE CODIFICACIÓN	116
2.5. MÓDULO DE BASES DE DATOS	117
3. REQUISITOS DE SOFTWARE Y HARDWARE	118
3.1. REQUERIMIENTOS MÍNIMOS DE HARDWARE	118
3.2. REQUERIMIENTOS MÍNIMOS DE SOFTWARE.....	118
4. APLICATIVO CALISOFT.....	119
4.1. ICONOS GENERALES.....	119
4.2. INGRESO AL APLICATIVO.....	119
4.2.1. Página principal.....	120
4.2.2. Inicio de sesión.....	121
4.2.3. Registro	122
4.3. USO DEL APLICATIVO	124
4.3.1. Modificar datos personales y cerrar sesión	124
4.3.2. Notificaciones.....	127
4.3.3. Administrador	129
4.3.4. Evaluador	140
4.3.5. Desarrollador	168
5. CONTROL DE CAMBIOS DEL MANUAL	198

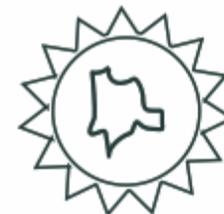


TABLA DE IMÁGENES

Imagen 1 Página principal	120
Imagen 2 Componentes página principal	120
Imagen 3 Inicio sesión.....	121
Imagen 4 Enlace recuperar contraseña	122
Imagen 5 Vista recuperar contraseña.....	122
Imagen 6 Enlace de registro.....	123
Imagen 7 Vista de registro de usuario	123
Imagen 8 Opciones generales.....	124
Imagen 9 Ver datos de usuario y cerrar sesión.....	125
Imagen 10 Información personal.....	125
Imagen 11 Cambiar nombre y correo	126
Imagen 12 Cambiar foto de usuario	126
Imagen 13 Modificar contraseña.....	127
Imagen 14 Icono de notificaciones	128
Imagen 15 Estructura de la sección de notificaciones.....	128
Imagen 16 Vista de notificaciones	129
Imagen 17 Dashboard Administrador.....	130
Imagen 18 Crud Usuarios.....	130
Imagen 19 Modal crear usuario	131
Imagen 20 Administración de proyectos	132
Imagen 21 Datos generales de los proyectos	132
Imagen 22 Modal asignar evaluador	133



Imagen 23 Evaluadores asignados	133
Imagen 24 Vista administración de documentos	134
Imagen 25 Vista de componentes	135
Imagen 26 Modal crear componente	135
Imagen 27 Administración de semilleros y grupos de investigación	136
Imagen 28 Sección de porcentajes Administrador	137
Imagen 29 Vista categorías administrador	137
Imagen 30 Modal crear categorías	138
Imagen 31 Porcentajes bases de datos.....	139
Imagen 32 Porcentajes de codificación	140
Imagen 33 Dashboard evaluador	141
Imagen 34 Porcentajes de categorías Evaluador	141
Imagen 35 Porcentajes de codificación evaluador	142
Imagen 36 Porcentajes de base de datos evaluador	143
Imagen 37 Documentación de los estándares de codificación.....	143
Imagen 38 Componente de proyecto asignado evaluador.....	145
Imagen 39 Proyecto listo para evaluación, evaluador.....	146
Imagen 40 Dropdown List módulos evaluativos.....	146
Imagen 41 Módulo evaluativo de documentación, evaluador	147
Imagen 42 Sub menú de los documentos.....	147
Imagen 43 Calificación de documentos	148
Imagen 44 Vista de módulo de plataforma, evaluador	148
Imagen 45 Modal crear caso prueba	149
Imagen 46 Vista de módulo de plataforma con caso prueba cread, evaluador	150
Imagen 47 Detalles caso prueba	151



Imagen 48 Caso prueba listo para calificar, evaluador	152
Imagen 49 Módulo de Testing, evaluador	153
Imagen 50 Proceso de Testing	154
Imagen 51 Proceso de testeo finalizado	155
Imagen 52 Tabla de Archivos de codificación del evaluador	156
Imagen 53 Modal ayuda de lista de Scripts	156
Imagen 54 Panel de calificación de Scripts	157
Imagen 55 Modal de ayuda para Calificar un Script	158
Imagen 56 Ejemplo Block UI	159
Imagen 57 Panel de Calificación Manual	159
Imagen 58 Panel de Calificación Manual con Resultados.....	160
Imagen 59 Panel de comentarios	160
Imagen 60 Diferencia entre Script calificado y sin calificar	161
Imagen 61 Tabla de Ítems sin Evaluar Evaluador	161
Imagen 62 Tabla de ítems Evaluados.....	162
Imagen 63 Reporte del módulo para la evaluación de codificación Evaluador.....	163
Imagen 64 Nota final calculando el índice de cohesión o aceptación evaluador.....	163
Imagen 65 Proceso final de calificación de proyectos.....	168
Imagen 66 Dashboard del desarrollador	169
Imagen 67 Registro del proyecto.....	170
Imagen 68 Proyecto listo para enviarlo como propuesta.....	171
Imagen 69 Modal invitar usuario.....	171
Imagen 70 Usuarios invitados.....	172
Imagen 71 Aceptar invitación	172
Imagen 72 Sección de integrantes actualizada.....	173

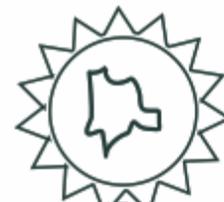


Imagen 73 Proyecto listo para evaluación.....	174
Imagen 74 Vista de documentación, desarrollador.....	175
Imagen 75 Modal para subir documentación.....	175
Imagen 76 Opciones de la documentación.....	176
Imagen 77 Primera Vista del módulo de codificación.....	176
Imagen 78 Modal de ayuda para subir archivos de codificación.....	177
Imagen 79 Panel de carga para subir Archivos de codificación.....	177
Imagen 80 Panel de Carga con Archivos de codificación.....	178
Imagen 81 tabla de listado de archivos de codificación estudiante.....	179
Imagen 82 Modal para eliminar un Script.....	179
Imagen 83 Panel de Visualización del código del desarrollador.....	180
Imagen 84 Proyecto en estado de evaluación.....	186
Imagen 85 Vista de resultados de evaluación de modelado, desarrollador.....	187
Imagen 86 Vista caso pruebas, desarrollador.....	188
Imagen 87 Ubicación de la extensión de Calisoft.....	189
Imagen 88 Estructura de la extensión de Calisoft.....	189
Imagen 89 Obtención de atributos de un formulario.....	190
Imagen 90 Modal para enviar caso prueba. desarrollador.....	191
Imagen 91 Resultados de la calificación de los casos de prueba, desarrollador.....	192
Imagen 92 Modal detalles de la evaluación del caso prueba.....	193
Imagen 93 Reporte de calificación del caso prueba.....	194
Imagen 94 Panel de Archivos de codificación calificados en la vista del desarrollador.....	195
Imagen 95 Tabla de Ítems sin Evaluar en la vista del desarrollador.....	195
Imagen 96 Tabla de Ítems evaluados en la vista del desarrollador.....	196
Imagen 97 Reporte del módulo para la evaluación de codificación Desarrollador.....	197

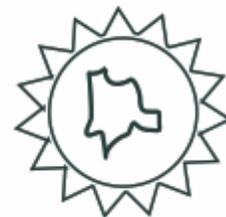


Imagen 98 Nota final calculando el índice de cohesión o aceptación Desarrollador 197



TABLA DE TABLAS

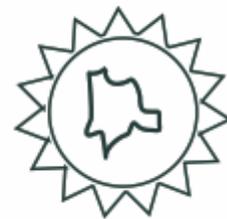
Tabla 1 Iconos generales.....	119
-------------------------------	-----



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



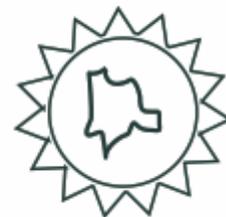
diferentes campos, con el objetivo de verificar la calidad en la seguridad al momento de ingresar datos erróneos o maliciosos al sistema, este procedimiento se realiza mediante el módulo de plataforma.

- 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 creó que el 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.

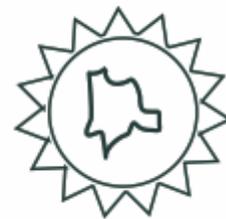
2.4. MÓDULO DE CODIFICACIÓN

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 encargará 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 indicará 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

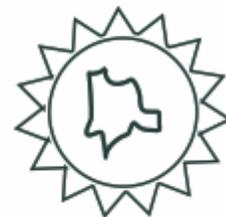
Memoria RAM: Mínimo: 1 Gigabytes (GB)

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

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

Tabla 30 Iconos generales

4.2. INGRESO AL APLICATIVO

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

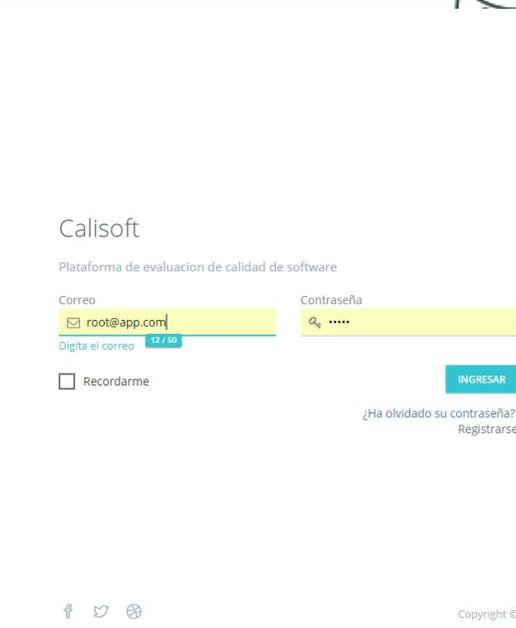


Imagen 23 Página principal

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;

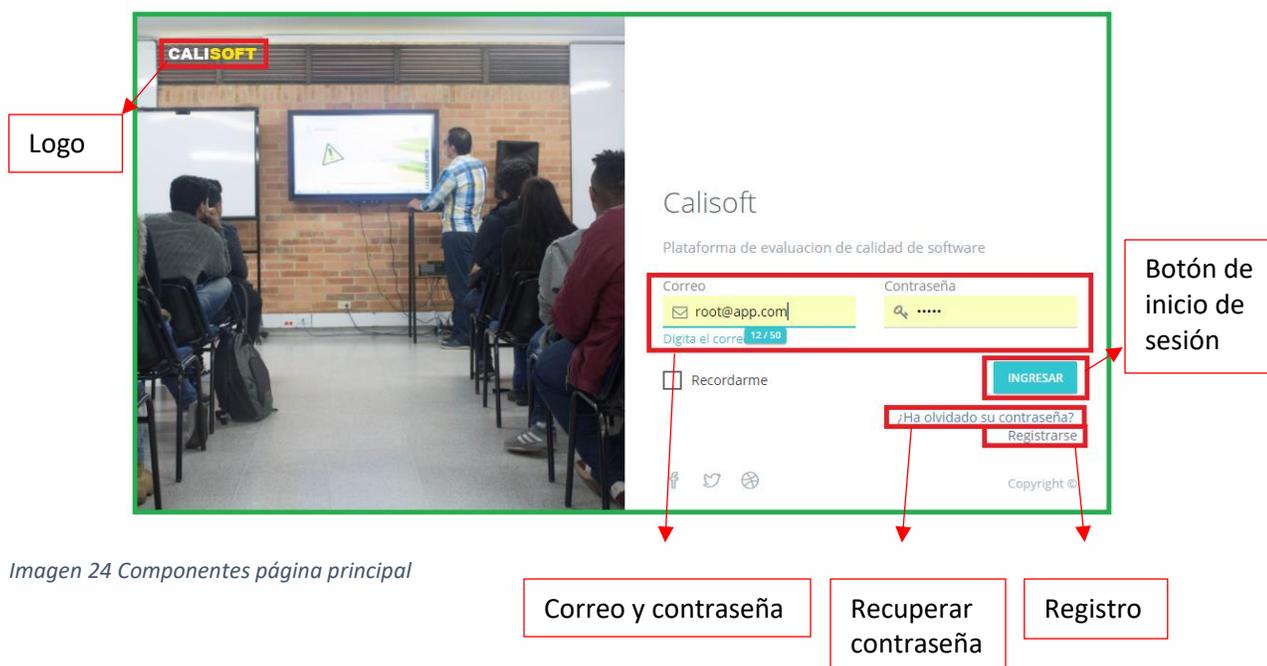
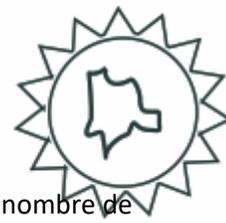


Imagen 24 Componentes página principal



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;

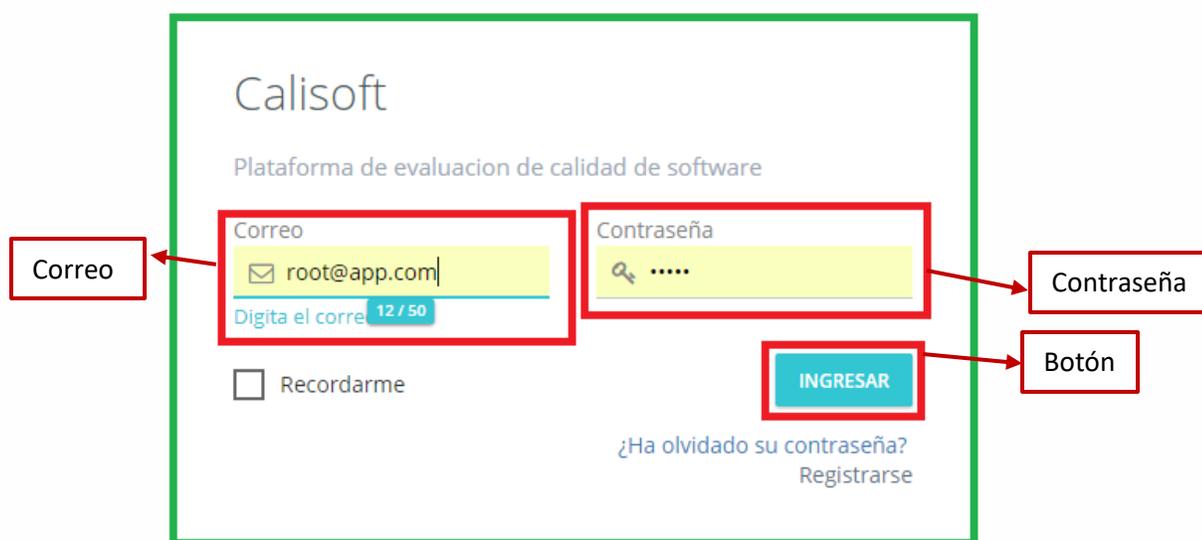


Imagen 25 Inicio sesión

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?”.



Calisoft

Plataforma de evaluación de calidad de software

Correo Digita el correo 12 / 50

Contraseña

Recordarme

[¿Ha olvidado su contraseña?](#)

INGRESAR

Recuperar contraseña

Imagen 26 Enlace recuperar contraseña

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

Calisoft

Plataforma de evaluación de calidad de software

Ha olvidado su contraseña ?

Introduzca su dirección de correo electrónico, a continuación para restablecer su contraseña.

Correo

CANCELAR

ENVIAR

Correo correspondiente a la cuenta que desea recuperar la contraseña

Botón

Imagen 27 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"



Calisoft

Plataforma de evaluación de calidad de software

Correo Contraseña

Digita el correo 12 / 50

Recordarme

¿Ha olvidado su contraseña?
[Registrarse](#)

Enlace hacia el registro

Imagen 28 Enlace de registro

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

Calisoft

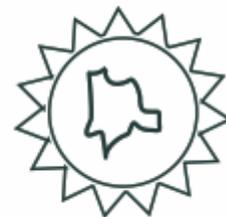
Plataforma de evaluación de calidad de software

Email Nombre

Contraseña Confirmar Contraseña

Imagen 29 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 contara con tres usuarios creados por defecto pertenecientes a los tres roles que maneja el aplicativo. El administrado 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. Modificar datos personales y cerrar sesión

Todos los usuarios cuentan con un apartado donde pueden modificar sus datos personales, al igual que la contraseña y cambiar su foto de perfil. Para acceder a esta sección, debe situarse en la esquina superior derecha

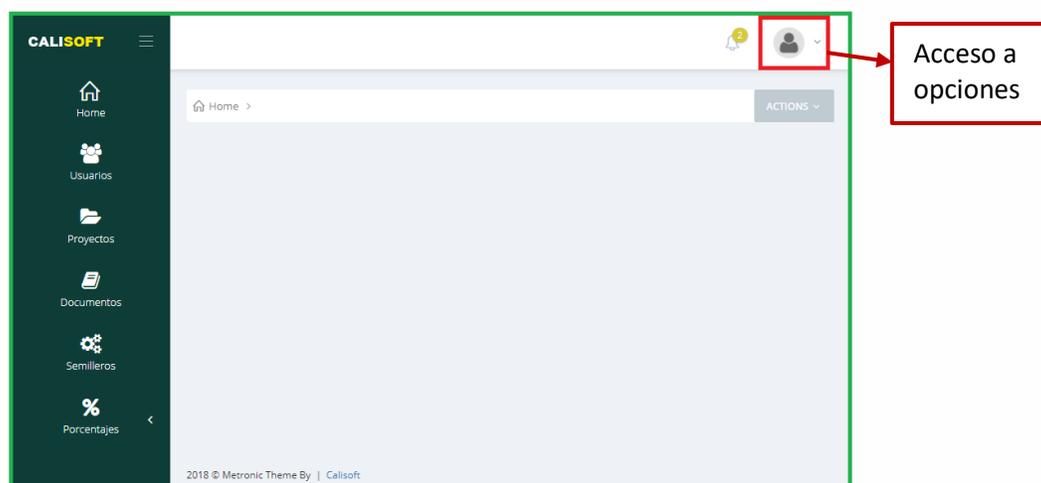


Imagen 30 Opciones generales

En donde deberá dar clic en la flecha o en la foto de perfil y se le desplegará un Dropdown List con dos opciones;

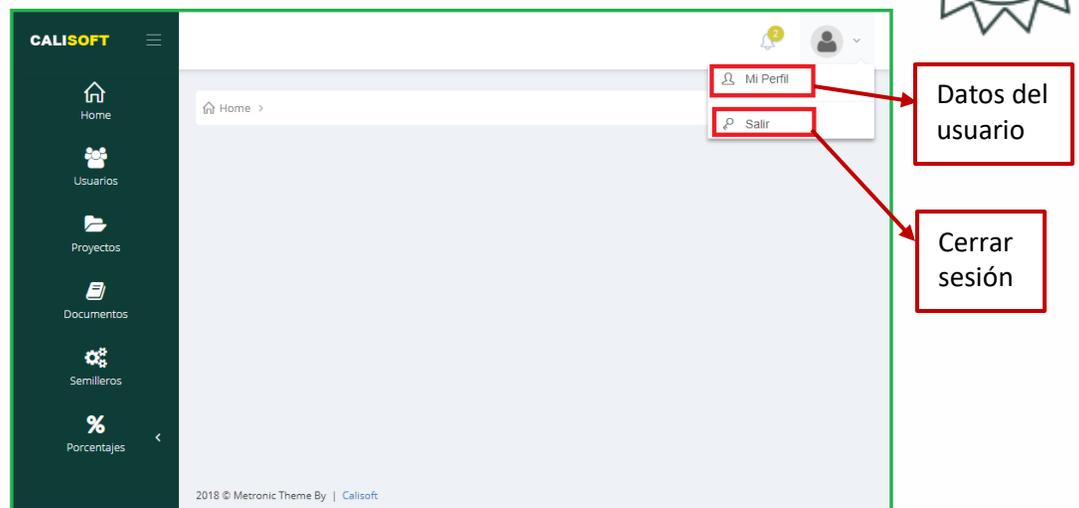


Imagen 31 Ver datos de usuario y cerrar sesión

La primera nos redirigirá a una vista en donde se pueden ver y modificar los datos personales del usuario, la segunda es la opción para cerrar sesión. Al dar clic en la primera opción, nos mostrará la siguiente vista;

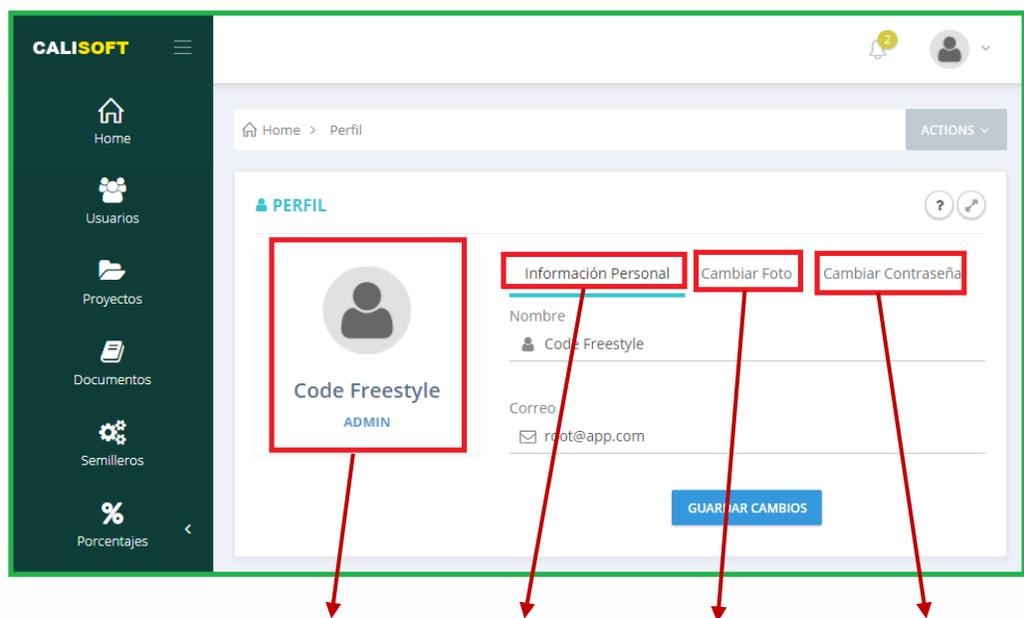


Imagen 32 Información personal

- Foto de usuario
- Nombre de usuario y correo
- Cambiar foto de usuario
- Modificar contraseña

La información correspondiente al nombre de perfil y correo se pueden modificar en la primera pestaña nombrada "Información Personal"

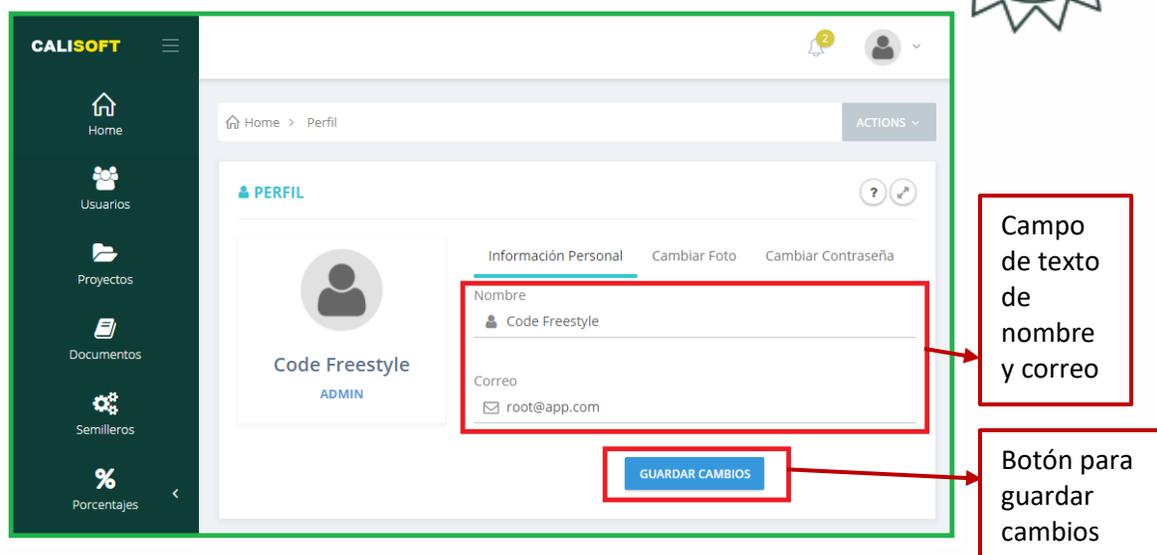


Imagen 33 Cambiar nombre y correo

En el campo de texto correspondiente a “Nombre” edite el nombre de usuario por el cual quiere que los demás usuarios lo vean y reconozcan, en el campo “Correo” edite el correo al cuál le llegarán las notificaciones e información general. Una vez haya modificado los campos de texto, oprima el botón “GUARDAR CAMBIOS” para modificar sus datos en la base de datos.

Al seleccionar la pestaña “Cambiar Foto” la sección de abajo cambiará y mostrará la siguiente opción;

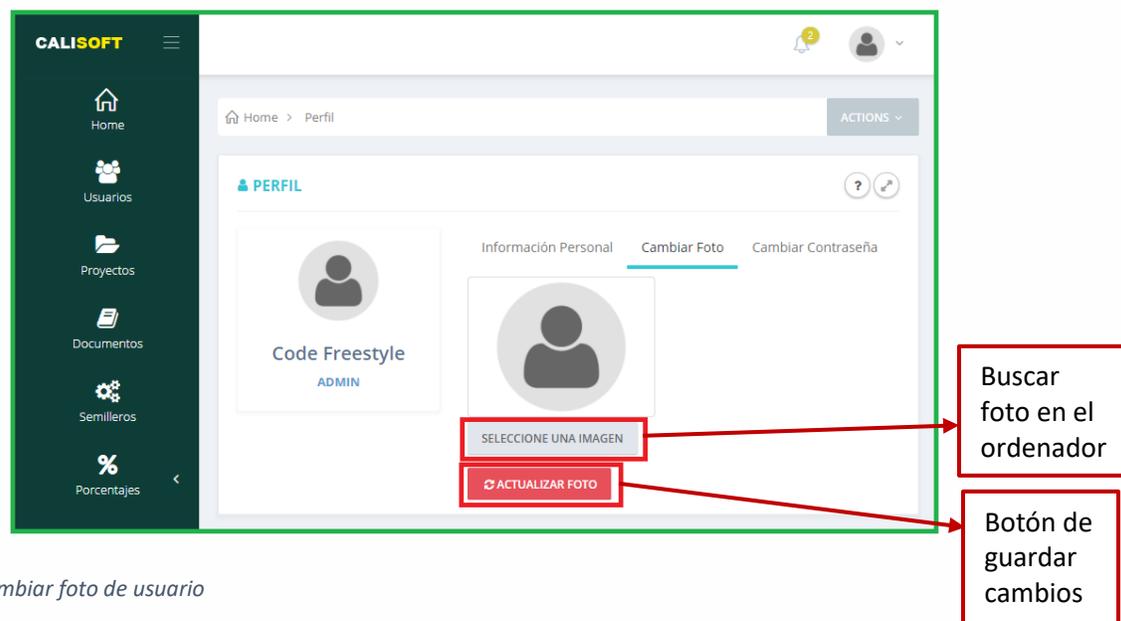


Imagen 34 Cambiar foto de usuario



Por medio del botón “SELECCIONE UNA IMAGEN” usted podrá buscar una imagen dentro de su ordenador, al seleccionar la imagen se previsualizará y si es de su agrado puede oprimir en el botón “ACTUALIZAR FOTO” para guardar los cambios.

Por último, al seleccionar la tercera pestaña “Cambiar contraseña” la sección de abajo cambiará y mostrará las siguientes opciones;

Imagen 35 Modificar contraseña

En el primer campo debe ingresar la contraseña que actualmente tenga registrada en su cuenta, una vez digitada la contraseña anterior, proceda a digitar una contraseña nueva en el segundo campo, la misma contraseña nueva se debe repetir en el tercer campo, de clic en el botón “EDITAR CONTRASEÑA” para guardar los cambios. En esta última etapa finalizamos todo lo referente a modificar datos personales.

4.3.2. Notificaciones

Las notificaciones en la aplicación ayudan a dar aviso de ciertos cambios que realizan los usuarios dentro de la plataforma, cambios como; creación de proyectos, asignaciones de proyectos, eliminación de proyectos, calificación, invitaciones, etc. Este apartado se encuentra en la parte superior derecha con un símbolo de campana;

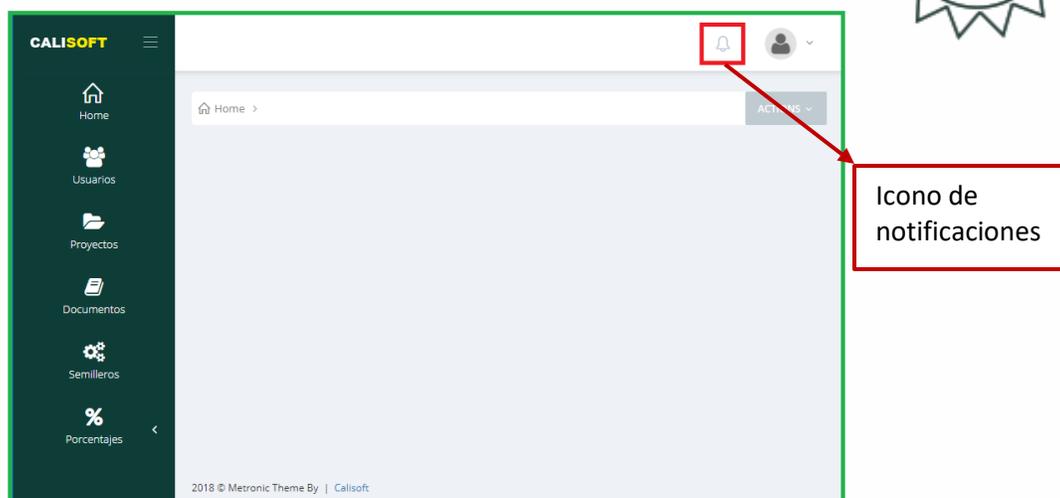
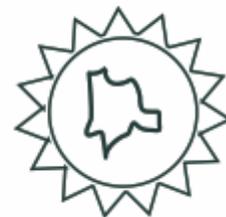


Imagen 36 Icono de notificaciones

En tiempo real, cuando ocurra una acción que conlleve a una notificación, sobre el icono aparecerá un número que irá acorde al número de notificaciones sin revisar que el usuario tiene. Al dar clic sobre el icono se desplegará el siguiente menú;

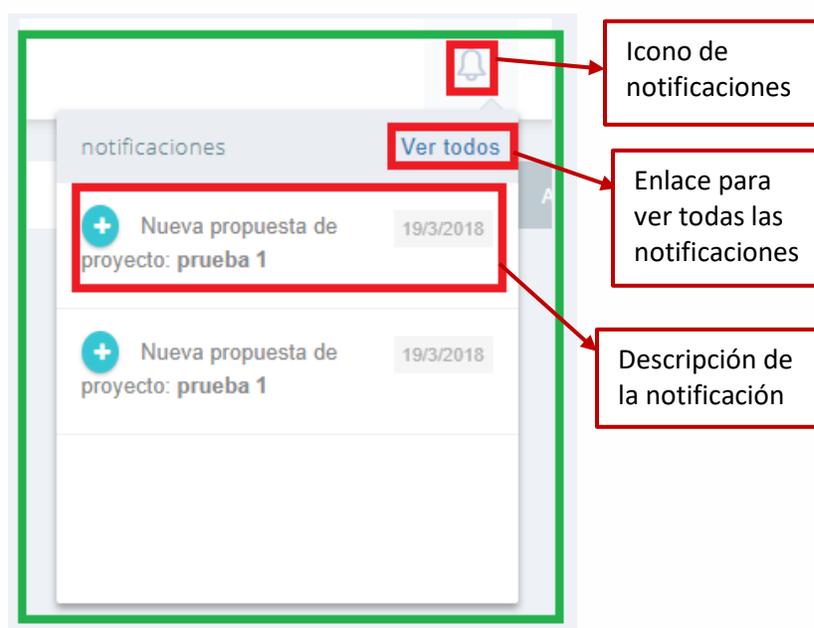


Imagen 37 Estructura de la sección de notificaciones

Al volver dar clic en la campana, la sección se minimizará, el enlace “Ver todos” redirige a una vista donde se podrá ver de manera más detallada las notificaciones, la descripción de las notificaciones también es un enlace que redirige a la vista o módulo donde pertenezca esa notificación, al darle clic en el enlace “Ver todos” redirigirá a la siguiente vista;

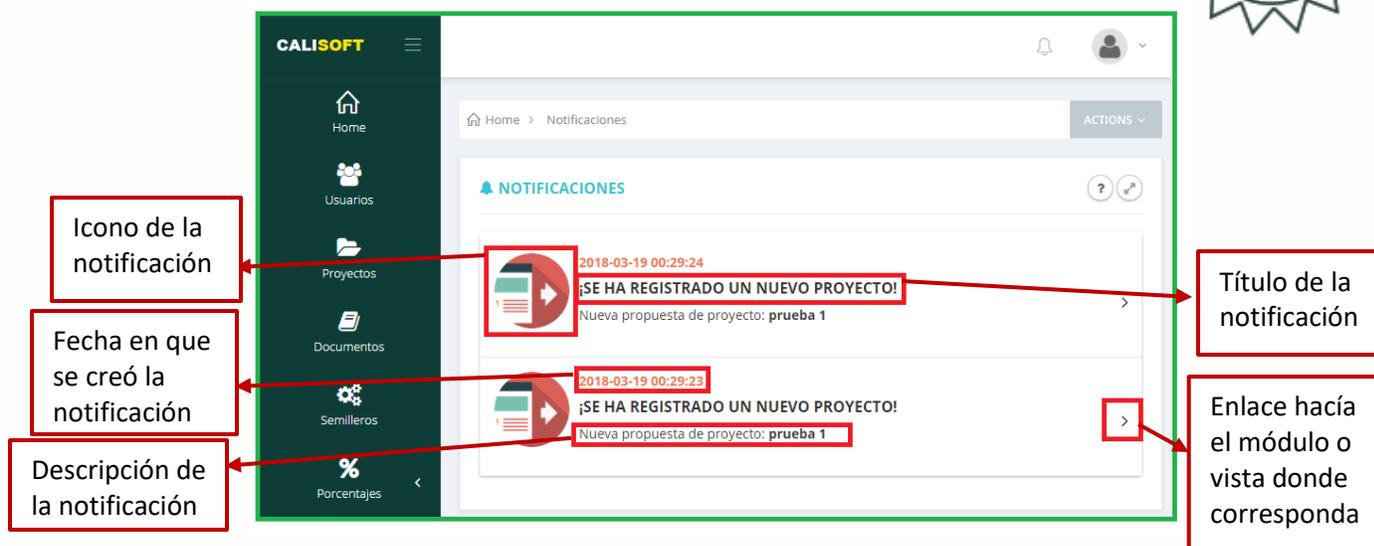


Imagen 38 Vista de notificaciones

Acá se podrá ver en orden cronológico las diferentes notificaciones que lleguen a nuestra bandeja, además, algunas notificaciones cuentan con una descripción más amplia, se recomienda entrar y leer cuidadosamente cada una de ellas.

4.3.3. Administrador

El administrador es el usuario principal, encargado de crear otros usuarios administradores o evaluadores, es el encargado de dar los porcentajes de calificación de los diferentes módulos, aceptar o denegar proyectos, asignar evaluadores y entre otras más funciones. En este apartado se hablará de las funciones que puede realizar el usuario con rol administrativo en la plataforma.

La siguiente imagen muestra la estructura del dashboard del administrador;

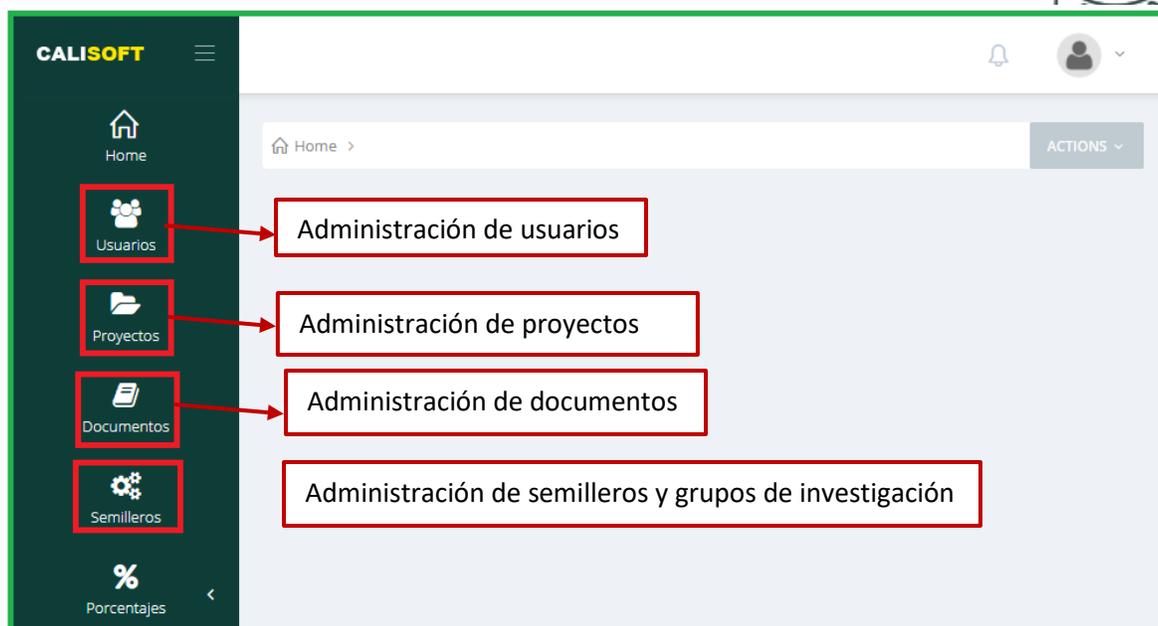


Imagen 39 Dashboard Administrador

Administración de usuario: el administrador podrá crear dos roles de usuario que son; administrador y evaluador, también puede optar por eliminar usuarios pertenecientes a los tres roles. La imagen a continuación corresponde a la vista de administración de usuarios;

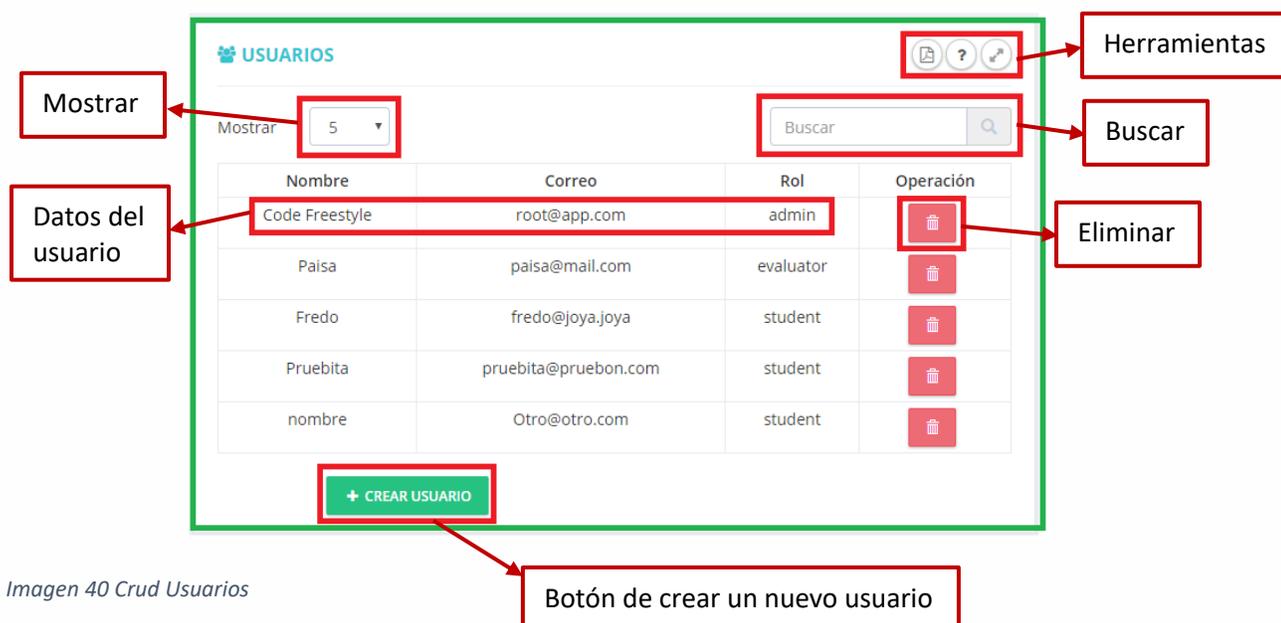


Imagen 40 Crud Usuarios

La sección de herramientas permite exportar el contenido actual de la tabla a un formato PDF, también cuenta con una ayuda que orientará al usuario en las diferentes funciones y una opción para agrandar la ventana hasta los marcos del navegador.



La sección de buscar permite, valga la redundancia, buscar un carácter o una palabra dentro de la tabla actual, esta búsqueda se realiza en tiempo real.

La sección de mostrar es una ayuda cuando la tabla cuenta con múltiples registros y al momento de listar esa gran cantidad de registros se torna tedioso, esta sección nos permitirá listar únicamente el número de registros que se encuentre seleccionado.

La sección de datos del usuario corresponde a los datos de nombre, correo y rol de los diferentes usuarios registrados en la base de datos, el administrador podrá eliminar los usuarios que desee.

Al presionar el botón de “CREAR USUARIO” se desplegará un modal que servirá para crear un nuevo usuario, la imagen a continuación corresponde a dicho modal;

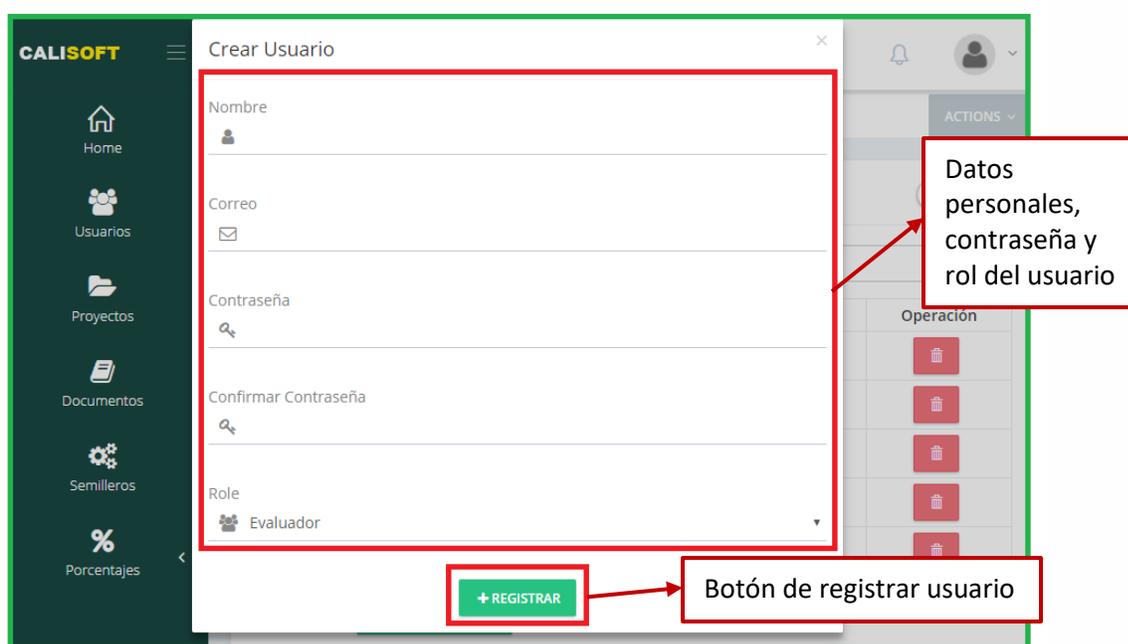


Imagen 41 Modal crear usuario

En el modal se puede apreciar un formulario que corresponde a los datos personales del usuario, además de la contraseña con la cual iniciará sesión y el rol al que pertenecerá, al momento de pulsar el botón “REGISTRAR” se guardará el usuario en la base de datos y se enviara automáticamente un mensaje con el contenido de la cuenta al correo con el que se creó el usuario.

Administración de proyectos: el administrador puede observar todos los proyectos que se encuentren creados, dependiendo del proyecto, el administrador tiene la voluntad de aceptar o no el proyecto, una vez aceptado deberá asignar un evaluador a ese proyecto. La imagen a continuación corresponde a la vista de administración de proyectos;

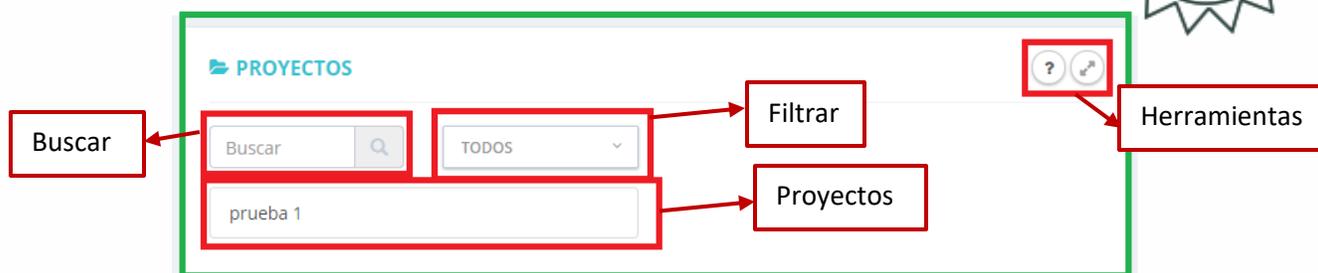
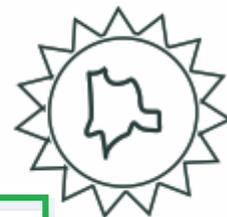


Imagen 42 Administración de proyectos

La sección de herramientas aporta ayuda al usuario, orientándolo por las diferentes funciones y también permite ampliar el componente hasta las esquinas del navegador.

La sección de filtrar permite agrupar los proyectos por tres estados, que son; todos, activo y propuesta. De esta forma se le facilitará al administrador seleccionar los proyectos que deba aceptar o realizar alguna modificación. La sección de buscar ayuda en un propósito similar, filtrando por algún carácter o palabra los proyectos que se encuentren en la base de datos.

LA sección de proyectos corresponde a los proyectos que los estudiantes hayan enviado como propuesta, además, es un componente que al dar clic se desplegarán las características del proyecto, como se puede ver en la imagen a continuación;

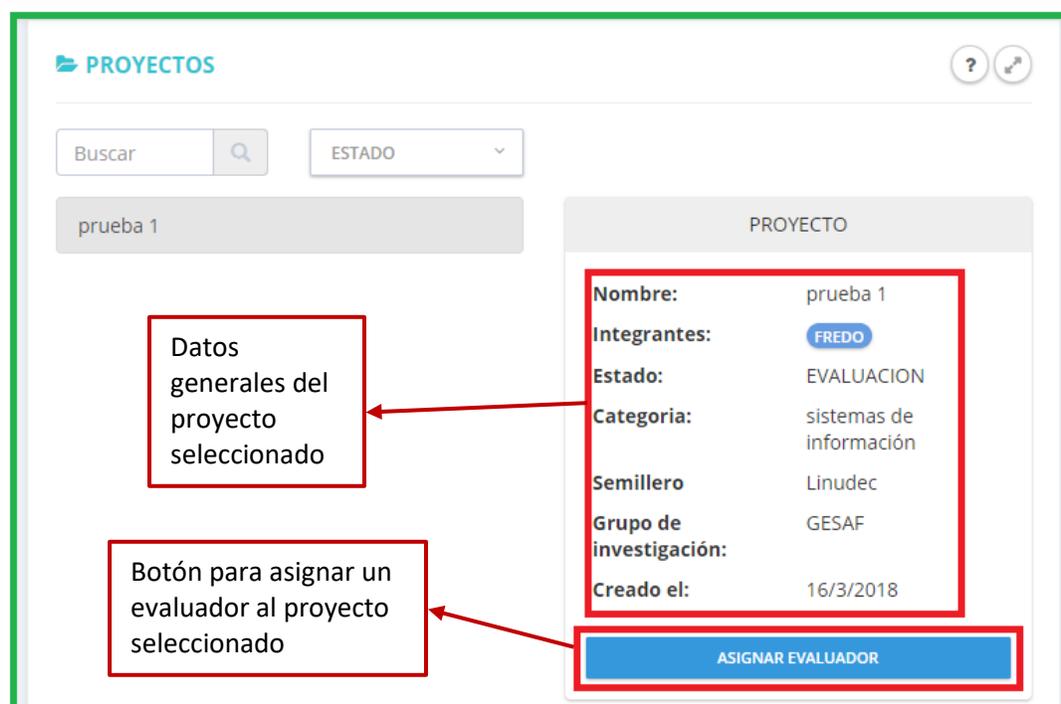


Imagen 43 Datos generales de los proyectos



En los datos generales del proyecto podemos encontrar el nombre completo del proyecto, los integrantes desarrolladores del proyecto y evaluadores si se le han asignado, el estado en el que se encuentra, que puede ser en creación, propuesta, activo, evaluación o completado, la categoría, semillero y grupo de investigación a la cual pertenece, y la fecha en que se creó. Al dar clic en el botón “ASIGNAR EVALUADOR” se desplegará un modal con las siguientes características;

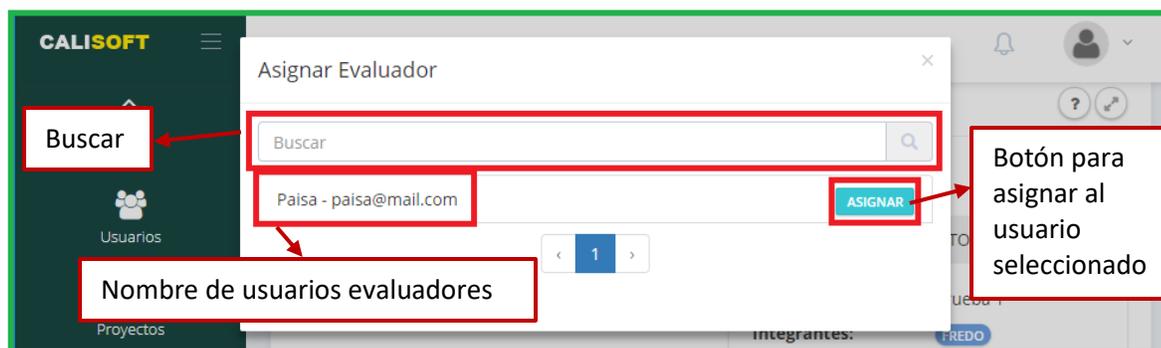


Imagen 44 Modal asignar evaluador

El componente de buscar ayuda a filtrar por algún carácter o palabra los evaluadores que se encuentren registrados en la base de datos, aparecerá el nombre del evaluador junto con su correo y al frente se sitúa un botón nombrado “ASIGNAR” que su función es seleccionar ese usuario como el evaluador del proyecto que se seleccionó anteriormente.

Al asignar al evaluador, se disparará una notificación a los integrantes del proyecto y al evaluador, además, en los datos del proyecto debe aparecer el nombre del evaluador como se ve en la siguiente imagen;

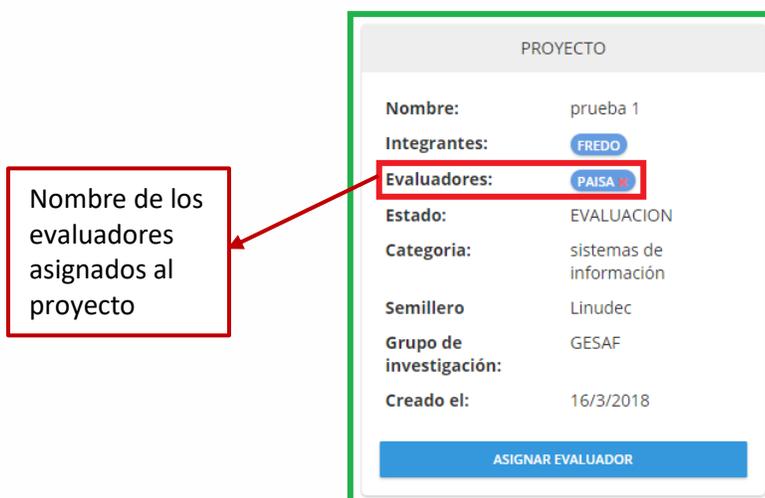


Imagen 45 Evaluadores asignados



Administración de documentos: el administrador puede crear los tipos de modelo que desee, al momento de crearlos puede escoger si el modelo es requerido o no, además también puede crear los componentes que pertenezcan a dicho modelo, y de igual forma, puede seleccionar que competentes deben ser requeridos. Cuenta con opciones de editar o eliminar modelos, al igual que editar y eliminar componentes. La imagen a continuación corresponde a la vista de administración de documentos;



Imagen 46 Vista administración de documentos

La vista cuenta con una sección de herramientas que ayudan al usuario a orientarse en las funciones de la vista y a expandirla hasta las esquinas del navegador.

Para crear un nuevo documento, debe situarse en la sección superior izquierda, el campo de texto con la etiqueta nombre corresponde al nombre del documento que desea crear, a la derecha de este encontrará un componente que define el si el documento a crear será obligatorio o no para el desarrollador. Una vez definido el nombre, deberá dar clic en el botón “Agregar” en la sección de abajo.

En la sección “nombres de los documentos creados” corresponde a todos los documentos que se hayan registrado en la base de datos, por defecto el sistema crea los que aprecia en la imagen anterior, la etiqueta de documento requerido señala que el documento es obligatorio para el desarrollador, en la parte de la derecha puede encontrar los botos de editar o eliminar el documento al cual corresponda la fila.

Para agregar componentes a los documentos creados, basta con dar clic en el botón “COMPONENTES” y el sistema nos redirigirá a la siguiente vista;



Nombre	Obligatorio	Descripción	Operaciones
Notación de clase	SI	Se usan para denotar el nombre de la clase atributos y operaciones	 
Interfaces	NO	Si se usa una interfaz se garantiza que las clases soporten un comportamiento requerido que permite que el sistema trate los elementos no relacionados en la misma manera	 
Tablas	SI	Una tabla es una clase estereotipada	 
Asoc		a que dos elementos del modelo tienen una instancia de un	 
Genera		Una generalización se usa p	 
Agregaciones	SI	Las agregaciones se usan para describir elementos que están compuestos de componentes más pequeños	 
Clase asociación	SI	Una clase asociación es una estructura que permite una conexión de asociación para tener conexiones y atributos	 
Dependencias	SI	Una dependencia se usa para modelar un alto rango de relaciones dependientes entre elementos del modelo	 
Trazado	SI	La relación de trazado es una especialización de una dependencia, vinculando elementos del modelo o conjuntos de elementos que representan la misma idea a través de los modelos	 
Relaciones	NO	Se usa para expresar trazabilidad e integridad en el modelo	 
Anidamientos	SI	Un anidamiento es un conector que muestra que el elemento fuer	 

Datos sobre los componentes creados

Editar o eliminar componente

Botón para crear un nuevo componente

+ CREAR NUEVO COMPONENTE

Imagen 47 Vista de componentes

Los datos sobre los componentes creados corresponden al nombre del componente, si es obligatorio o no para el desarrollador y una breve descripción que ayuda al desarrollador a tener una idea del componente por si no lo conoce. Al costado derecho puede encontrar las opciones de modificar o eliminar el componente que corresponda a la fila. Sobre los documentos que se crean por defecto, estos componentes también son creados automáticamente, si el administrador desea agregar un nuevo componente debe dirigirse a costado de abajo y dar clic en el botón “CREAR NUEVO COMPONENTE” y el sistema abrirá un modal con las siguientes características;

Crear Componentes

Nombre

OBLIGATORIO OFF requerido

Descripción

+ GUARDAR **CANCELAR**

Definir si el componente es obligatorio o no

Nombre del componente a crear

Descripción del componente a crear

Botón para crear el componente

Imagen 48 Modal crear componente



El modal cuenta con una sección en la parte superior para definir el nombre del componente, en la parte de abajo cuenta con la opción de definir si es obligatorio o no y más puede dar una descripción que ayuda al desarrollador a tener una idea más detallada del componente. Una vez llenado todos los campos debe proceder a dar clic en el botón “GUARDAR” para crear el componente, este se anexará al listado en tiempo real.

Administración de semilleros y grupos de investigación: El administrador puede registrar los semilleros o grupos de investigación que se encuentren estipulados en la Universidad de Cundinamarca, al igual que eliminar o editar los datos con los que se registraron. La imagen a continuación corresponde a la vista de administración de semilleros o grupos de investigación;

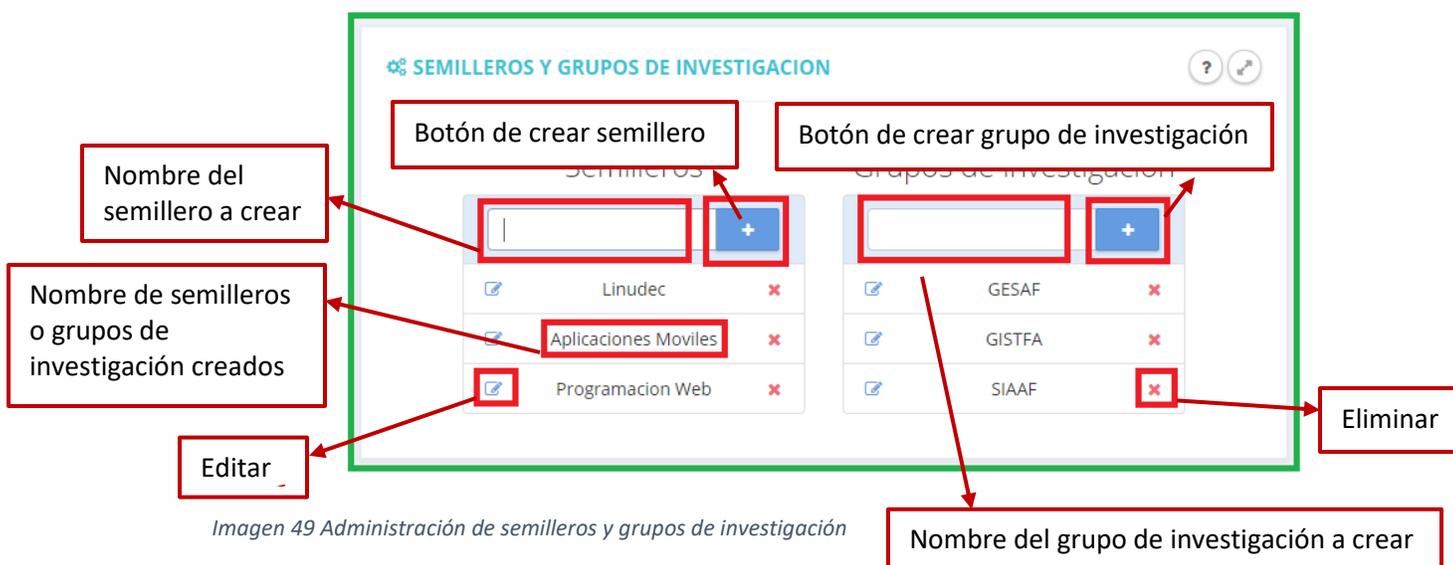


Imagen 49 Administración de semilleros y grupos de investigación

LA vista cuenta con dos cajas de texto que corresponden al nombre del semillero o grupo de investigación que se desea registrar, al lado derecho de cada caja de texto se ubica un botón con un icono de más que su acción es completar el registro del semillero o grupo de investigación, en la parte de abajo se enlistan los semillero y grupos de investigación que se encuentren registrados en la base de datos, al costado izquierdo de cada registro se encuentra un icono que sirve como enlace para modificar el nombre del registro, al costado derecho de cada registro se encuentra un icono con forma de equis que sirve como enlace para eliminar el registro seleccionado.

El apartado de abajo corresponde a los porcentajes, si se le da clic en el icono se desplegará un submenú correspondiente a los porcentajes de las categorías, bases de datos y codificación.

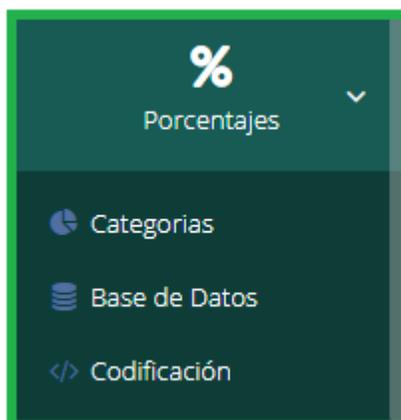
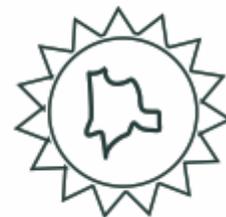


Imagen 50 Sección de porcentajes Administrador

Al darle clic en la sección de categorías aparecerá la siguiente vista;

Nombre	Plataforma	Modelado	Base de datos	Codificación	Operaciones
sistemas de información	25	25	25	25	[Icono de descripción] [Icono de editar] [Icono de eliminar]
sistemas de embebidos	25	25	25	25	[Icono de descripción] [Icono de editar] [Icono de eliminar]
sistema artificial	25	25	25	25	[Icono de descripción] [Icono de editar] [Icono de eliminar]

Nombre de la categoría y descripción de los porcentajes en los diferentes módulos de evaluación

Editar o eliminar categoría

Descripción

+ CREAR NUEVA CATEGORÍA

Botón de crear categoría

Imagen 51 Vista categorías administrador

La vista muestra la información de cada categoría que se encuentre registrada en la base de datos, los porcentajes corresponden a los diferentes módulos de evaluación, cabe resaltar que dependiendo de la categoría donde el desarrollador registre su proyecto, la evaluación se registrará bajo esos porcentajes. Al costado derecho se ubican las operaciones que se pueden hacer por cada registro, entre ellas tenemos la de editar y eliminar categoría, y un icono que despliega una descripción que será de ayuda para el buen manejo de cada categoría. Estas categorías son las que genera el sistema por defecto. En la parte de abajo se sitúa el botón "CREAR NUEVA CATEGORÍA" que al dar clic abrirá un modal para registrar una nueva categoría, la imagen a continuación corresponde a dicho modal;



The image shows a modal window titled "Crear Categoría" with a close button (X) in the top right corner. The form contains the following fields and elements:

- Nombre de la Categoría:** A text input field with a circular icon on the left. A red box highlights the input area, and an arrow points to a label box containing "Nombre de la categoría".
- Evaluación de plataforma:** A field showing "% 0".
- Evaluación de modelado:** A field showing "% 0".
- Evaluación de Base de datos:** A field showing "% 0".
- Evaluación de Codificación:** A field showing "% 0".
- Descripción:** A text area with a vertical cursor. A red box highlights the text area, and an arrow points to a label box containing "Descripción".
- Botón de crear categoría:** A label pointing to the green "+ CREAR CATEGORÍA" button.
- Botón de cancelar:** A red button labeled "CANCELAR".

A red box on the left side of the form groups the four percentage fields, with an arrow pointing to a label box containing "Porcentajes de los diferentes modulos de evaluación".

Imagen 52 Modal crear categorías

El modal cuenta en su parte superior una sección para definir el nombre de la categoría, en la parte de abajo se definen los porcentajes correspondientes a los cuatro módulos evaluativos de Calisoft, además de un apartado en donde se redacta una breve descripción sobre la categoría. Una vez llenado todos campos, se debe proceder a dar clic sobre el botón "CREAR CATEGORÍA" para crear y finalizar este proceso.

Si se selecciona la opción de bases de datos, aparecerá la siguiente vista;



BASE DE DATOS

Componente	Estandar	Nomenclatura	Valor	Operación
Base de Datos	BDS_		3	
Esquemas	SCH_		4	
Tablas	TBL_	CREATE TABLE `TBL_	4	
Vistas	VWS_		4	
Llaves Primarias	PK_	PRIMARY KEY (`PK_	5	
Llaves Foraneas	FK_	FOREIGN KEY (`FK_	5	
Campo Descripcion	PGS_		3	
Campo ValorMoneda	CTB_		3	
Campo Observaciones	PSN_		3	

Descripción de los componentes evaluativos

Editar registro

Imagen 53 Porcentajes bases de datos

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 evaluativos

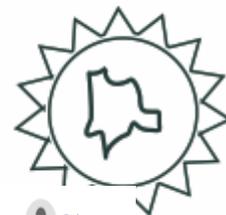
Editar registro

Imagen 54 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 ítems evaluativos y a su costado derecho cuenta con la opción de editar el registro correspondiente a la fila seleccionada.

4.3.4. 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 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 diferentes módulos

Sección informativa sobre los estándares de codificación y base de datos

Proyectos asignados

Imagen 55 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 “Categorías” dentro de la sección de porcentajes, el sistema nos redirigirá a la siguiente vista;

Nombre	Plataforma	Modelado	Base de datos	Codificación	Descripción
sistemas de información	25	25	25	25	
sistemas de embebidos	25	25	25	25	
sistema artificial	25	25	25	25	

Porcentajes de cada módulo conforme su categoría

Descripción de la categoría

Imagen 56 Porcentajes de categorías Evaluador



En donde el evaluador podrá ver la descripción de cada categoría junto con los porcentajes que corresponde a cada una.

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
Identacion	3
Comentarios	3
Espacios De Nombre	2

Imagen 57 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 “Base de Datos” dentro de la sección de porcentajes, el sistema nos redirigirá a la siguiente vista;

Componente	Estandar	Nomenclatura	Valor
Base de Datos	BDS_		3
Esquemas	SCH_		4
Tablas	TBL_	CREATE TABLE `TBL_	4
Vistas	VWS_		4
Llaves Primarias	PK_	PRIMARY KEY (`PK_	5
Llaves Foraneas	FK_	FOREIGN KEY (`FK_	5
Campo Descripcion	PGS_		3
Campo ValorMoneda	CTB_		3
Campo Observaciones	PSN_		3

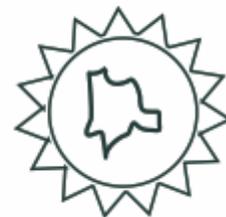


Imagen 58 Porcentajes de base de datos evaluador

En donde el evaluador podrá observar el valor que tiene cada componente de una base de datos, además de ver como debe ser su correcto estándar o nomenclatura.

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

</> 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

Secciones → 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 59 Documentación de los estándares de codificación

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

ESTANDAR DE BASE DE DATOS

Con este proyecto se busca desarrollar una herramienta para evaluar la calidad y modelación de las bases de datos, creadas por los estudiantes de ingeniería de sistemas de la universidad de Cundinamarca, con el fin de mejorar el uso, creación de la estructura y los componentes elaborados.

Esta herramienta facilitará los parámetros en los estándares aplicados para determinar el grado de las necesidades de los usuarios permitiendo evaluar el nivel en que los modelos correspondan a los datos que se desean almacenar.

El desarrollo está basado en la norma ISO/IEC 25012- ISO/IEC 19139 que presenta de forma detallada los modelos de calidad de software y datos en la cual establecen las características externas que a su vez se descomponen en sub-características y proporcionando consejos prácticos sobre el uso de modelos.

En el siguiente documento de la republica de Colombia, se presenta la forma adecuada de hacer uso del estandar, en el lineamiento para nombramiento de bases de datos. Con el objetivo de describir el desarrollo en las areas de informacion y de sistemas, tenido en cuenta los componentes que se evaluaran en los archivos sql.

VER ESTANDAR → Estandar Base de Datos

Imagen 60 Documentación de los estándares de Base de Datos



En la vista tendrá la debida información acerca de que herramientas se utilizan, las normas en las que está basada el estándar y hacia quien está desarrollado el proyecto. Además, se puede ver el estándar para implementar en la nomenclatura de las bases de datos.



Imagen 61 Documento de estándar de base de datos de la Presidencia de la Republica.

En esta vista el evaluador tendrá a su disposición el manual de uso del estándar de codificación, las palabras que aparecen de color azul corresponden a enlaces que al dar clic se actualizará la información de abajo. Se recomienda que el evaluador lea toda la documentación antes de realizar el proceso de calificación a un proyecto.

Al momento en que se le asigna un proyecto a un evaluador, en la página principal de su dashboard aparecerá el siguiente segmento;



PROYECTOS ASIGNADOS

prueba 1

Estado *El proyecto todavía no se encuentra listo para evaluación*

Nombre:	prueba 1
Estado:	ACTIVO
Categoría:	sistemas de información
Semillero	Linudec
Grupo de investigación:	GESAF
Creado el:	16/3/2018
Integrantes:	FREDO
Evaluadores:	PAISA

Información sobre el proyecto asignado

Imagen 62 Componente de proyecto asignado evaluador

Cada proyecto que se le asigne al evaluador se visualizará como se observa en la imagen anterior, el apartado cuenta con una sección de información del proyecto, en donde se puede ver el nombre del proyecto, estado en que se encuentra, a qué categoría pertenece, semillero o grupo de investigación al cual se registró, la fecha de creación, integrantes y los evaluadores que tenga asignado ese proyecto. En la parte superior cuenta con un apartado que corresponde a un estado, si está en la forma anterior significa que el desarrollador no ha subido la documentación de su proyecto y no ha cambiado el estado a “EVALUACIÓN”, una vez el desarrollador haga lo anterior descrito, el segmento aparecerá de la siguiente forma;



PROYECTOS ASIGNADOS

prueba 1 EVALUACIÓN ▾

Nombre:	prueba 1
Estado:	EVALUACION
Categoría:	sistemas de información
Semillero	Linudec
Grupo de investigación:	GESAF
Creado el:	16/3/2018
Integrantes:	FREDO
Evaluadores:	PAISA

VOLVER A ACTIVAR PROYECTO TERMINAR EVALUACIÓN

Información sobre el proyecto asignado

Botón para activar de nuevo el proyecto

Botón para terminar la evaluación

Imagen 63 Proyecto listo para evaluación, evaluador

Se habilita las opciones de evaluación que al dar clic se despliegan los cuatro módulos evaluativos, como se ve en la imagen a continuación;

MODELACIÓN

P PLATAFORMA

E CODIFICACIÓN

S BASES DE DATOS

Linudec

Imagen 64 Dropdown List módulos evaluativos

Si se escoge la opción de “MODELACIÓN” el sistema redirigirá al evaluador a la siguiente vista;

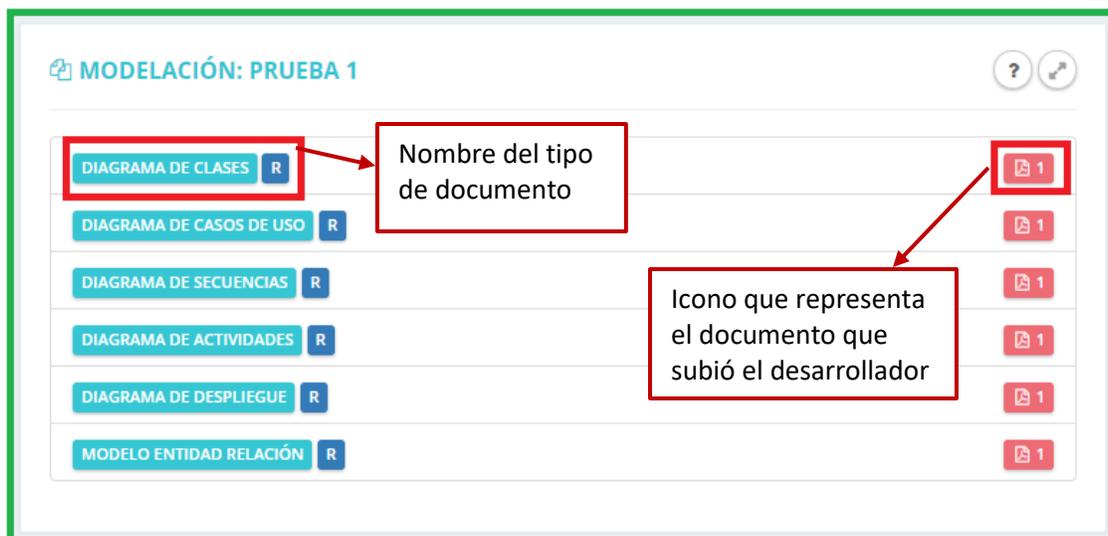


Imagen 65 Módulo evaluativo de documentación, evaluador

En donde el evaluador podrá observar los documentos que subió el desarrollador, si el evaluador da clic sobre los iconos que hay en cada fila correspondiente a los tipos de documentos, se desplegará el siguiente submenú;

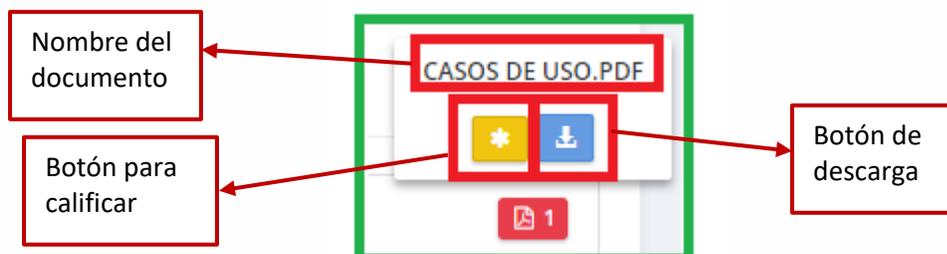
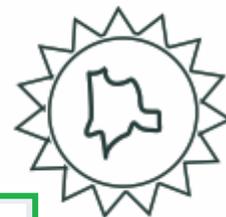


Imagen 66 Sub menú de los documentos

Este submenú cuenta con el nombre del documento que subió el desarrollador, y dos botones, el botón de la derecha descarga el documento y el botón de la izquierda nos redirigirá a la vista en donde se realiza la evaluación de ese documento. La vista de evaluación es la siguiente;



EVALUAR DOCUMENTO: DIAGRAMA DE CASOS DE USO

Componente desplegado

Visualización del documento

Actores

Los actores representan los roles que pueden incluir usuarios humanos, un hardware externo u otros sistemas

Descripción

Observación

Observación

Guardar calificación

NO VALIDO GUARDAR

Define si el componente es correcto o incorrecto

Componente sin desplegar

Casos de uso

Inclusión de casos de uso

Casos de uso extendidos

Puntos de extensión

Límite del sistema

Imagen 67 Calificación de documentos

La imagen anterior muestra la estructura de la vista de calificación de documentos, donde cuenta a la izquierda con una sección donde se puede visualizar el documento que subió el desarrollador, en la parte de la derecha, se encuentran los componentes que pertenecen a un tipo de documento, en este ejemplo, al tipo “Casos de uso”, al dar clic en los componentes se desplegará un sub menú que contiene una descripción del componente, un apartado de observación por si el evaluador quiere dejar una recomendación, un botón con dos estados que define si el componente está bien empleado o no, y un botón que guarda los cambios realizados.

Volviendo a la página principal del evaluador y en el apartado de proyectos se escoge la opción de “PLATAFORMA” como se ve en la imagen 42, el sistema redirigirá al evaluador a la siguiente vista;

PLATAFORMA: PROYECTO DE PRUEBA

+ CREAR CASO PRUEBA

Botón para crear un caso prueba

Imagen 68 Vista de módulo de plataforma, evaluador

Como inicio, el desarrollador debe dar clic en el botón “CREAR CASO PRUEBA” y el sistema abrirá el siguiente modal;



Nombre del caso prueba

Proposito del caso prueba

Alcance del caso prueba

Límite del caso prueba

Botón para crear el caso prueba

Resultado esperado del caso prueba

Criterios del caso prueba

Prioridad del caso prueba

Imagen 69 Modal crear caso prueba

Diligenciar el modal anterior es necesario para crear un caso prueba, el caso prueba debe corresponder a un caso de uso que el evaluador tenga en su documentación, el desarrollador debe ver en qué casos de uso hay procesos de entrada de datos al sistema y deberá crear el caso de prueba correspondiente, por ejemplo, un caso prueba del login o registro. La descripción para cada campo es la siguiente;

- **Nombre de la prueba:** cada caso de prueba debe tener un nombre que lo identifique, debe ser algo corto y entendible para el desarrollador.
- **Propósito:** cuando se crea un caso de prueba se debe decir el propósito, lo que se propone a realizar especificando un fin, también es un objetivo importante y amplio a largo plazo.
- **Alcance:** en el caso de prueba se debe especificar hasta donde llega el propósito, que es lo que se pretende evaluar y de acuerdo con esto el desarrollador podrá saber hasta dónde tiene que subir la prueba.
- **Resultados esperados:** en esta parte de las especificaciones el evaluador da entender al desarrollador que es lo que espera de la prueba, cuáles son los resultados que se debe obtener al finalizar la prueba.
- **Criterios a evaluar:** el desarrollador debe conocer además del propósito y los resultados esperados todos los criterios del evaluador, son cada una de las pruebas que se van a realizar al caso de prueba, así el desarrollador podrá subir la prueba mostrando estos



critérios, por ejemplo, validación de cajas de texto, que no halla inconsistencias con los requerimientos, y que la funcionalidad se cumpla entre otras.

- **Prioridad:** una especificación más es la prioridad, la prioridad es una configuración realizada por el administrador y el evaluador tiene tres opciones para clasificarla según los requerimientos, estos tres tipos son prioridad alta, media y baja, de esta manera les da más importancia o mayor valor a unos casos de prueba respecto a otros.
- **Fecha límite:** cada caso de prueba tiene un tiempo mínimo para poder subir la prueba, si esa fecha se incumple y se vence el plazo de subir la prueba, el desarrollador ya no tendrá acceso a subir la prueba, en ese caso el evaluador tiene la opción de modificar la fecha, pero ya quedaría a su propio criterio.

Una vez diligenciado el documento, el evaluador debe dar clic en el botón “CREAR” y la vista de plataforma se actualizará de la siguiente manera;

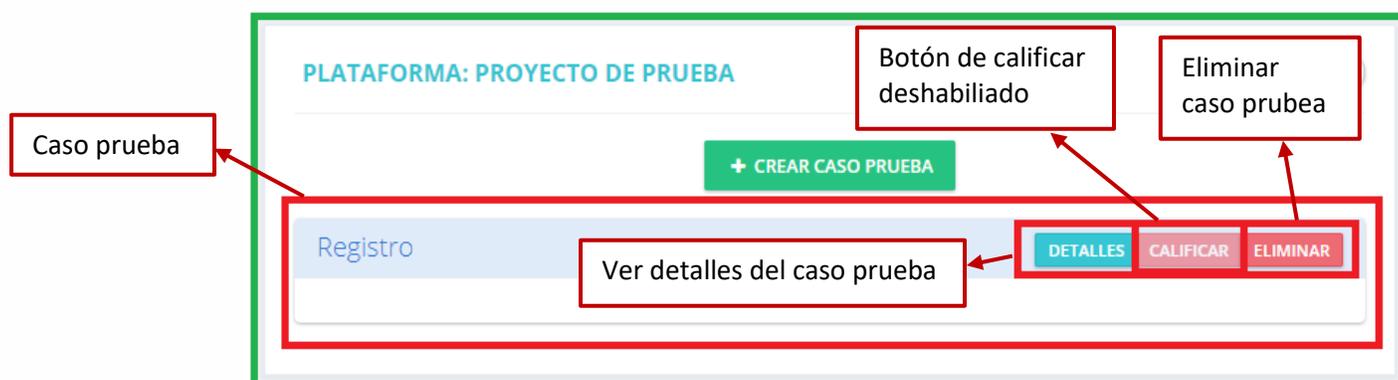


Imagen 70 Vista de módulo de plataforma con caso prueba cread, evaluador

El caso prueba anteriormente creado se muestra con tres opciones, el botón de “ELIMINAR” borra el caso prueba creado, el botón de “CALIFICAR” se encuentra deshabilitado hasta que el desarrollador suba el correspondiente formulario, y botón de “DETALLES”, si se oprime el botón se desplegará una sección que corresponde a los detalles del caso prueba, dicha sección es la siguiente;



PLATAFORMA: PROYECTO DE PRUEBA

+ CREAR CASO PRUEBA

Registro

DETALLES CALIFICAR ELIMINAR

Proposito:	Registrar un usuario.
Alcance:	Diligenciar el formulario de registro y enviarlo.
Resultado Esperado:	Registrar un usuario con daos coherentes.
Criterios:	El nombre no puede permitir números ni símbolos, además el correo debe ser coherente y debe estar a prueba de ataques.
Prioridad:	alta
Estado:	evaluar
Plazo:	2018-03-27 00:00:00

EL ESTUDIANTE NO A SUBIDO EL CASO PRUEBA

Etiqueta informativa

Detalles del caso prueba

Imagen 71 Detalles caso prueba

Desplegando de manera ordenada los detalles del caso prueba junto con una etiqueta que informa que aún el desarrollador no ha subido el correspondiente formulario. La creación del caso prueba es notificada al desarrollador, de esta forma el desarrollador puede subir el formulario correspondiente de forma oportuna, una vez subido el JSON correspondiente al formulario, el caso prueba se actualiza de la siguiente manera;



PLATAFORMA: PROYECTO DE PRUEBA

+ CREAR CASO PRUEBA

Registro

DETALLE CALIFICAR ELIMINAR

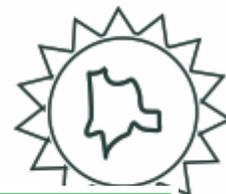
Proposito:	Registrar un usuario.
Alcance:	Diligenciar el formulario de registro y enviarlo.
Resultado Esperado:	Registrar un usuario con daos coherentes.
Criterios:	El nombre no puede permitir números ni símbolos, además el correo debe ser coherente y debe estar a prueba de ataques.
Prioridad:	alta
Estado:	evaluar
Plazo:	2018-03-30 20:20:09
Json subido:	Total de Inputs: 4

Botón de calificar habilitado

Información de inputs dentro del JSON

Imagen 72 Caso prueba listo para calificar, evaluador

El botón de calificación se habilitará y aparecerá un apartado donde informa la cantidad de inputs que subió el desarrollador, al dar clic sobre el botón "CALIFICAR" el sistema redirigirá al evaluador a la siguiente vista;



ESENARIO PRUEBA: REGISTRO ? ↻

0 👍
BARRA DE PROCESO
PORCENTAJE 0%

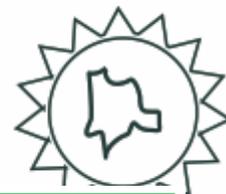
Tipo	Nombre	Reglas	Input	Estado
email	email	email required max:50	<input type="text"/>	✘
text	name	required	<input type="text"/>	✘
password	password	required	<input type="password"/>	✘
password	password_confirmation	required	<input type="password"/>	✘

+ CARGAR PRUEBA Botón para poner en marcha la prueba

Detalles del input

Imagen 73 Módulo de Testing, evaluador

El evaluador en este módulo dispondrá de una barra de proceso, que le indica las veces que debe realizar la prueba, debajo de la barra se disponen de la simulación de los inputs que se encontraron en el formulario en tipo JSON que subió el desarrollador, y al dar clic en el botón “CARGAR PRUEBA” el sistema automáticamente llenará los espacios en la columna Input y en estado saldrá si el campo cumple con las reglas del sistema, se debe ver de la siguiente manera;



ESCENARIO PRUEBA: REGISTRO

0
BARRA DE PROCESO

Estado que informa que los inputs están validados correctamente

PORCENTAJE 0%

Tipo	Nombre	Reglas	Input	Estado
email	email	email required max:50	oserna@yahoo.com	✓
text	name	required	dolore	✓
password	password	required	✓
password	password_confirmation	required	✓

Calificación obtenida

CALIFICACIÓN 100%

+ CARGAR PRUEBA

Guardar prueba

GUARDAR PRUEBA

Imagen 74 Proceso de Testing

Es necesario que, tras la operación de cargar prueba, el evaluador de clic en el botón para guardar la prueba que realizó, este proceso lo debe repetir hasta que la barra de proceso se llene completamente. Al finalizar, la vista se actualizará de la siguiente manera;



ESCENARIO PRUEBA: REGISTRO ? ↗

30 👍
BARRA DE PROCESO

PORCENTAJE 100%

Tipo	Nombre	Reglas	Input	Estado
email	email	email required max:50	<input type="text"/>	✘
text	name	required	<input type="text"/>	✘
password	password	required	<input type="password"/>	✘
password	password_confirmation	required	<input type="password"/>	✘

Imagen 75 Proceso de testeo finalizado

La opción de cargar prueba desaparece y el porcentaje llega a su tope que es 100%. En este punto el desarrollador será notificado y podrá ver los resultados en su dashboard.

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 42, el sistema redirigirá al evaluador a la siguiente vista;



Imagen 76 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 desplegaran 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 77 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

Calificación automática

Modal De ayuda

VALUAR DOCUMENTO: 6290_USERCONTROLLER.PHP

```
1 <?php
2 namespace App\Container\Calisoft\Src\Controllers;
3
4 use Illuminate\Http\Request;
5 use App\Http\Controllers\Controller;
6 use App\Container\Calisoft\Src\User;
7 use App\Container\Calisoft\Src\Repositories\Users;
8 use App\Container\Calisoft\Src\Requests\UserIndexRequest;
9 use App\Container\Calisoft\Src\Requests\UserStoreRequest;
10 use App\Container\Calisoft\Src\Requests\UserSearchEvaluatorsRequest;
11 use App\Container\Calisoft\Src\Requests\UserSearchFreeStudentsRequest;
12 use App\Container\Calisoft\Src\Notifications\UsuarioCreado;
13
14 class UserController extends Controller
15 /
```

Visualización del código fuente

Crear Comentario

Calificación Manual

Calificación automática

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

Comentario

ACEPTAR CALIFICACIÓN

Aceptar Calificación

Imagen 78 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 79 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

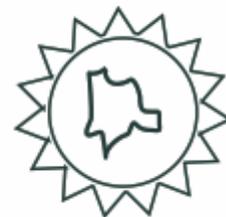


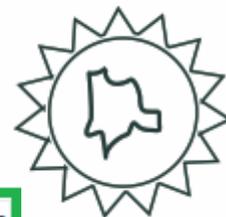
Imagen 80 Ejemplo Block UI

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 81 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 calificación
Constantes	0		0
Identacion	13	11	0.85
Comentarios	14	5	0.36
Espacios De Nombre	1	0	0

Imagen 82 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

Boton de aceptar calificacion y guardar comentario

Imagen 83 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 último, 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	
6568_TiposDocumentoController.php	
3567_TestInputsController.php	
3671_TestingController.php	

Script calificado

Script sin calificar

Imagen 84 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 85 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

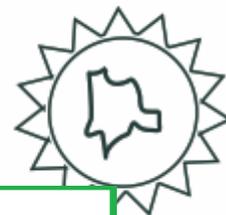


UserController.php			
Item	Atributo	Fila	Aprobado
Variables	users	16	✓
Variables	request	30	✓
Variables	user	44	✓
Variables	usuario	60	✓
Clases	UserController	14	✓
Funciones	index	30	✓
Funciones	store	41	✓
Funciones	destroy	60	✓
Funciones	proyectos	68	✓
Funciones	searchFreeStudents	82	✓
Funciones	searchEvaluators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

Imagen 86 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



UDECA
UNIVERSIDAD DE
CUNDINAMARCA

CALISOFT

Calle 14 con Avenida 15
Universidad de Cundinamarca - Ext. Facativá
(+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 87 Reporte del módulo para la evaluación de codificación Evaluador

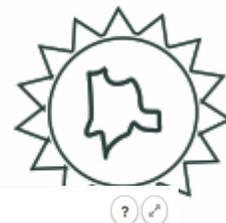
el reporte generado por todos los scripts, con las observaciones del evaluador, al final del reporte se muestra el porcentaje del índice de cohesión o aceptación que tiene el proyecto frente al módulo para la evaluación de estándares de codificación

Promedio General (Codificación)

71.5%

Imagen 88 Nota final calculando el índice de cohesión o aceptación evaluador

Volviendo a la página principal del evaluador y en el apartado de proyectos se escoge la opción de "BASES DE DATOS" como se ve en la imagen 42, el sistema redirigirá al evaluador a la siguiente vista;



NOMENCLATURA: GESTION DE CALIDAD

Codigo SQL

```
23 DROP TABLE IF EXISTS `TBL_Carpark_Dependencias`;  
24 CREATE TABLE `TBL_Carpark_Dependencias` (  
25   `PK_CD_IdDependencia` int(10) UNSIGNED NOT NULL,  
26   `CD_Dependencia` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_u  
27   `created_at` timestamp(0) NULL DEFAULT NULL,  
28   `updated_at` timestamp(0) NULL DEFAULT NULL,  
29   PRIMARY KEY (`PK_CD_IdDependencia`) USING BTREE  
30 ) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4_unicode_ci  
31  
32 -----  
33 -- Records of TBL_Carpark_Dependencias  
34 -----  
35 INSERT INTO `TBL_Carpark_Dependencias` VALUES (1, 'Administrativo', '2018-01-01', '2018-01-01', '2018-01-01', '2018-01-01');  
36 INSERT INTO `TBL_Carpark_Dependencias` VALUES (2, 'Profesor(a)', '2018-01-01', '2018-01-01', '2018-01-01', '2018-01-01');  
37
```

CALIFICACIÓN AUTOMÁTICA

Información			
Componente	Totales	Aprobados	Calificación
Base de Datos	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Esquemas	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Tablas	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Vistas	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Llaves Primarias	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Llaves Foraneas	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Campo Descripcion	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Campo ValorMoneda	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Campo Observaciones	<input type="text" value="0"/>	<input type="text" value="0"/>	0

Calificación Total: NAN

La vista general del módulo de base de datos se compone en, un modal de ayuda, en un visor de código SQL, en un botón que hará la calificación automática sin antes preguntar si se está seguro de hacer dicha calificación y un apartado donde sale una información general de los componentes, los totales, los aprobados y su respectiva calificación, donde sobresalen unos inputs donde el evaluador podrá hacer la calificación manual

NOMENCLATURA: GESTION DE CALIDAD

Codigo SQL

```
23 DROP TABLE IF EXISTS `TBL_Carpark_Dependencias`;  
24 CREATE TABLE `TBL_Carpark_Dependencias` (  
25   `PK_CD_IdDependencia` int(10) UNSIGNED NOT NULL,  
26   `CD_Dependencia` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_u  
27   `created_at` timestamp(0) NULL DEFAULT NULL,  
28   `updated_at` timestamp(0) NULL DEFAULT NULL,  
29   PRIMARY KEY (`PK_CD_IdDependencia`) USING BTREE  
30 ) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4_unicode_ci  
31  
32 -----  
33 -- Records of TBL_Carpark_Dependencias  
34 -----  
35 INSERT INTO `TBL_Carpark_Dependencias` VALUES (1, 'Administrativo', '2018-01-01', '2018-01-01', '2018-01-01', '2018-01-01');  
36 INSERT INTO `TBL_Carpark_Dependencias` VALUES (2, 'Profesor(a)', '2018-01-01', '2018-01-01', '2018-01-01', '2018-01-01');  
37
```

Ayuda Visualización Codigo SQL

Calificación Manual

Calificación Automática

CALIFICACIÓN AUTOMÁTICA

Modal Ayuda

Información			
Componente	Totales	Aprobados	Calificación
Base de Datos	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Esquemas	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Tablas	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Vistas	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Llaves Primarias	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Llaves Foraneas	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Campo Descripcion	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Campo ValorMoneda	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Campo Observaciones	<input type="text" value="0"/>	<input type="text" value="0"/>	0

Calificación Total: NAN

El modal de ayuda guiara al evaluador en los pasos de cada calificación.



Ayuda Módulo Calificar Archivo Base de Datos

Funciones Módulo Calificar Archivo Base de Datos

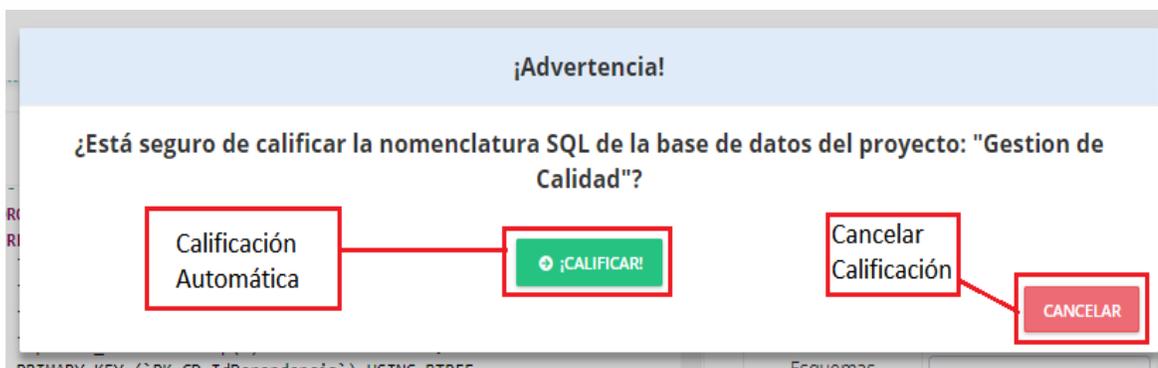
- Visualizar el archivo SQL para realizar la calificación.
- Asignar el numero total de componentes encontrados en el archivo SQL.
- Asignar el numero de componentes que se encuentran bien, segun el estandar establecido para la calificación adecuada.
- Automaticamente se realizara la calificacion de cada componente y su nota total del archivo.
- Podra agregar una observación al final de la calificación.

Modal Calificar Archivo

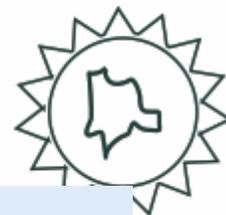
Cerrar Modal

CERRAR

El botón de calificación automática arroja un modal que le dice al evaluador, si está seguro de hacer la calificación automática, con el respectivo botón de calificar y con el botón de cancelar.



Al ejecutar la calificación automática nos saldrá un modal con la calificación de cada ítem del código SQL, mediado por los estándares dados por el mismo evaluador o administrador del software, los resultados de cada ítem son guardados en la base de datos para que consiguientemente, el software genere la calificación correspondiente del código SQL.



Calificación Nomenclatura Base de datos - Proyecto: "Gestion de Calidad"

CALIFICACIÓN AUTOMÁTICA ↕

Palabras Reservadas SQL	Total	Palabras Encontradas	Total
CREATE DATABASE	0	BDS_	0
CREATE SCHEMA	0	SCH_	0
CREATE TABLE	7	CREATE TABLE `TBL_	6
VIEWS	0	VWS_	0
PRIMARY KEY	7	PRIMARY KEY (`PK_	6
FOREIGN KEY	3	FOREIGN KEY (`FK_	3
PGS_	0	PGS_	0
CTB_	0	CTB_	0
PSN_	0	PSN_	0
Total Palabras Reservadas del SQL:	17	Total Palabras Encontradas del SQL:	15

¡Advertencia!

El proyecto: "Gestion de Calidad", Fue calificado correctamente.

Proyecto Calificado
Correctamente

¡ACEPTAR!

Si no se generó ningún error, saldrá un modal dando una advertencia, el cual dirá que el proyecto fue calificado correctamente.



Información

Componente	Totales	Aprobados	Calificación
Base de Datos	0	0	0
Esquemas	0	0	0
Tablas	7	6	4.29
Vistas	0	0	0
Llaves Primarias	7	6	4.29
Llaves Foraneas	3	3	5.00
Campo Descripción	0	0	0
Campo ValorMoneda	0	0	0
Campo Observaciones	0	0	0

Calificación Total: 4.53

Por otra parte, en la calificación manual el evaluador, podrá editar cada input o caja de texto, apoyándose del visualizador de código, para generar la calificación automática correcta. Sin embargo, cualquier calificación que seleccione el evaluador es válida, cada calificación llegara a una misma conclusión, teniendo en cuenta que es más confiable realizar la calificación automática.

La calificación total que genero el software, es la suma de la calificación de cada item, teniendo en cuenta que cada item tiene un valor diferente, dependiendo el estandar que se este manejando en ese momento.

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

Habilitar para correcciones

Terminar evaluación

Imagen 89 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.

4.3.5. 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 90 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 izquierda 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;



REGISTRO PROYECTO

Nombre del proyecto registrar

Nombre del proyecto

Categoría en la que se encuentra el proyecto

Categoría

sistemas de información

Semillero en el que se encuentra el proyecto

Semillero

Linudec

Grupo de investigación en el que se encuentra el proyecto

Grupo de investigación

GESAF

Botón para registrar el proyecto

ENVIAR

Imagen 91 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 92 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;

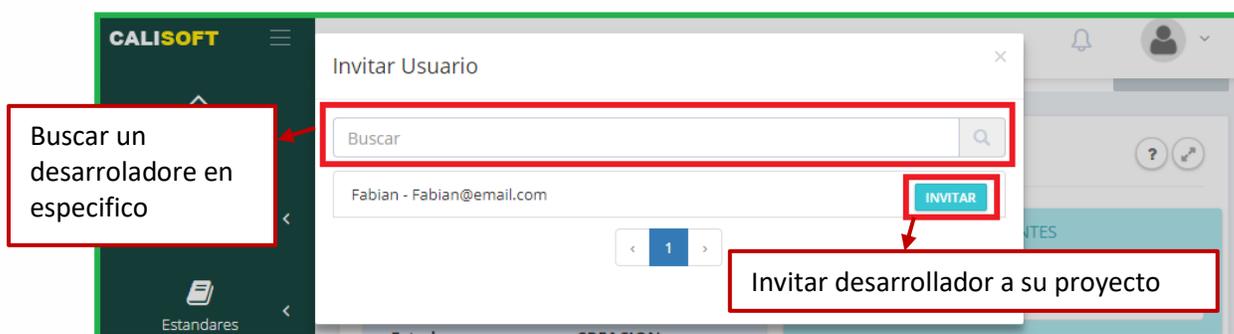


Imagen 93 Modal invitar usuario

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 project. On the left, under the heading 'PROYECTO', 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 'INTEGRANTES', the user 'johan - johan@flix.co' is listed. Below that, under 'INVITADOS', the user 'Fabian - Fabian@email.com' is listed with a '+ Invitar' button. A red box highlights the 'INVITADOS' section, and an arrow points from it to a separate box labeled 'Usuarios invitados'.

Imagen 94 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'. A message states: 'El proyecto Proyecto de prueba, a enviado una solicitud, para que haga parte del equipo de trabajo.' To the right of the message are two buttons: 'ACEPTAR' (blue) and 'RECHAZAR' (grey). A red box labeled 'Aceptar invitación al proyecto' has an arrow pointing to the 'ACEPTAR' button. Another red box labeled 'Rechazar invitación' has an arrow pointing to the 'RECHAZAR' button.

Imagen 95 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 page 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:

This section is highlighted with a red box and contains a list of members:

- Fabian - Fabian@email.com
- johan - johan@flix.co

INVITADOS Section:

This section is highlighted with a grey box and contains a button labeled "+ Invitar".

A red arrow points from the "INTEGRANTES" section to a text box that reads "Sección de integrantes actualizada".

Imagen 96 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;



Apartado para subir documentación del proyecto

Apartado para modelar la base de datos del proyecto

Botón para cambiar el estado del proyecto a "Evaluación"

Evaluadores asignados

Imagen 97 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 diagramas de su proyecto, para ello debe ir a la sección de documentos y dar clic en "Modelación" y el sistema lo redirigirá a la siguiente vista;

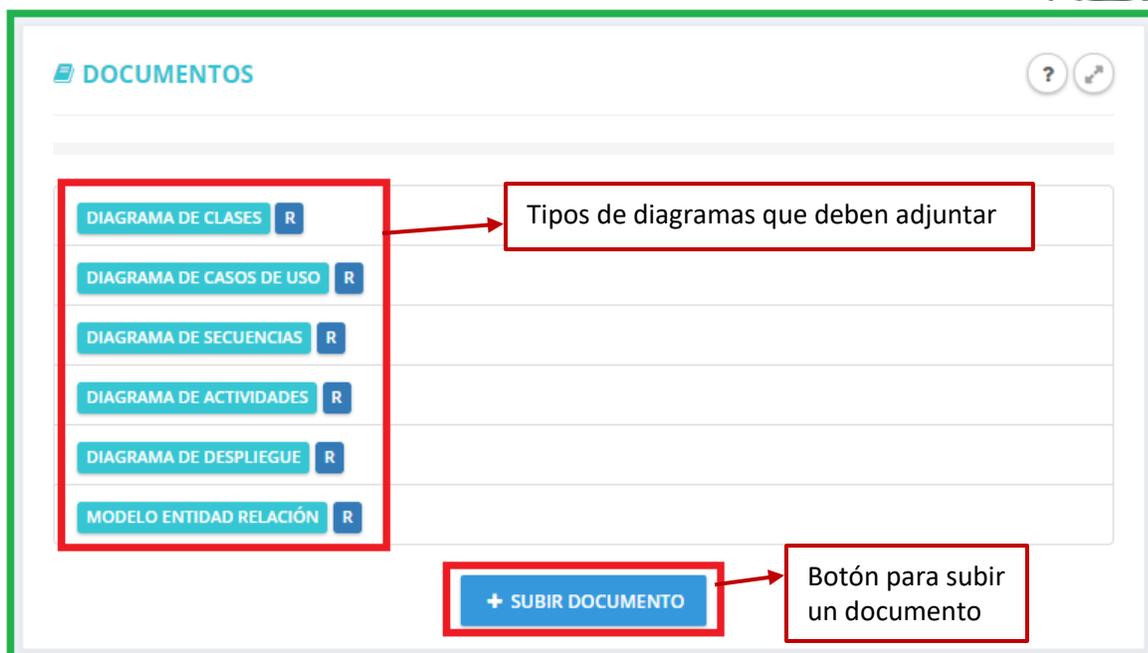


Imagen 98 Vista de documentación, desarrollador

La vista mostrará los tipos de documentos que debe subir el desarrollador y en la parte inferior se sitúa un botón que al oprimirlo desplegará un modal que será necesario para subir la documentación, la vista del modal es la siguiente;

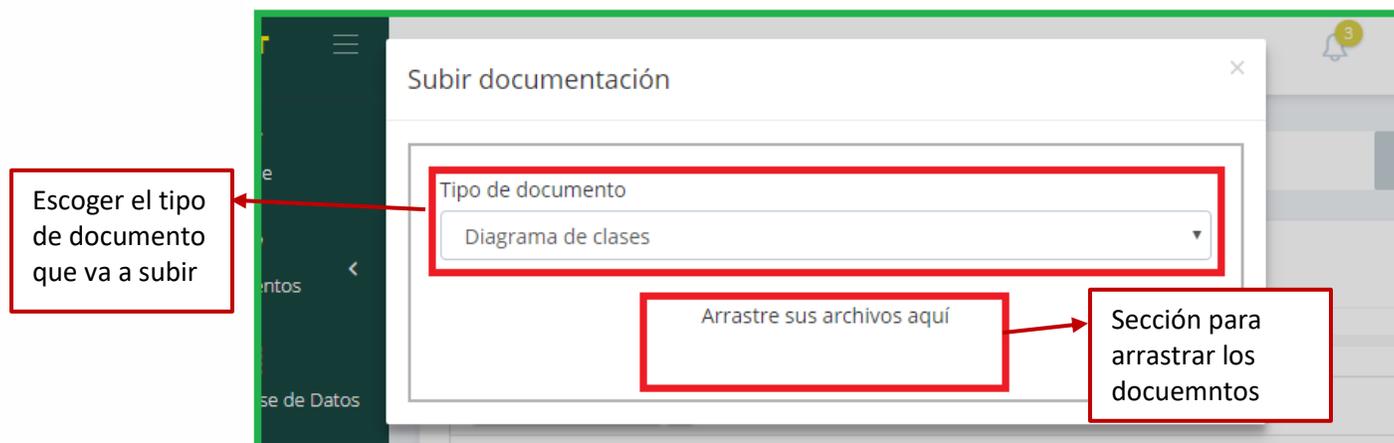


Imagen 99 Modal para subir documentación

El modal cuenta con dos secciones, una para escoger el tipo del documento que va a subir, y una segunda sección que por la cual usted podrá subir la documentación a la plataforma, sólo basta que arrastre el documento en formato "Pdf." Y el sistema automáticamente subirá a la base de datos el documento, sólo tenga en cuenta que antes de arrastrar el documento debe seleccionar el tipo al que corresponda. Una vez subido un documento la vista se actualizará de la siguiente forma;



Imagen 100 Opciones de la documentación

Aparecerá una barra de progreso en la parte superior que una vez llena, significará que el desarrollador ha subido todos los documentos requeridos para el sistema, en cada fila de los tipos de documentos aparecerá un icono que representa a un documento que subió el desarrollador, y al dar clic sobre él, se desplegará un menú con tres botones, uno para editar el documento, otra para eliminar el documento y el tercero para descargar el documento.

Como segundo 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

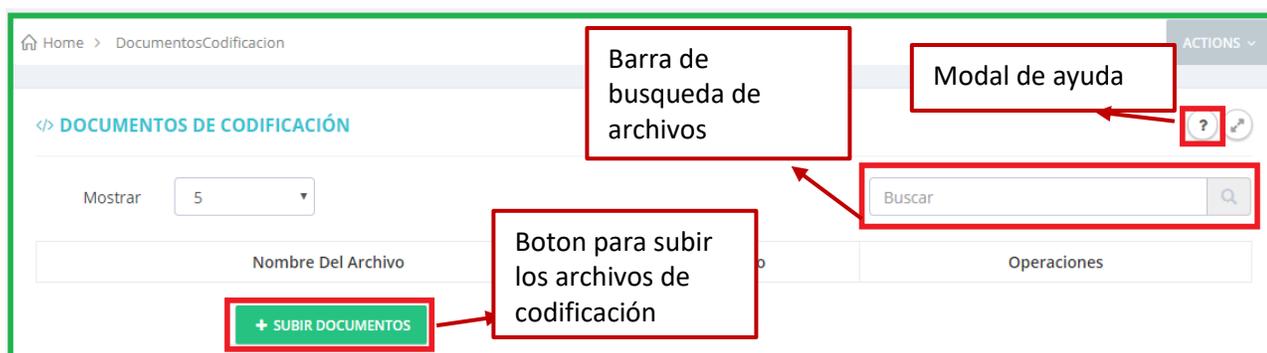


Imagen 101 Primera Vista del módulo de codificación



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

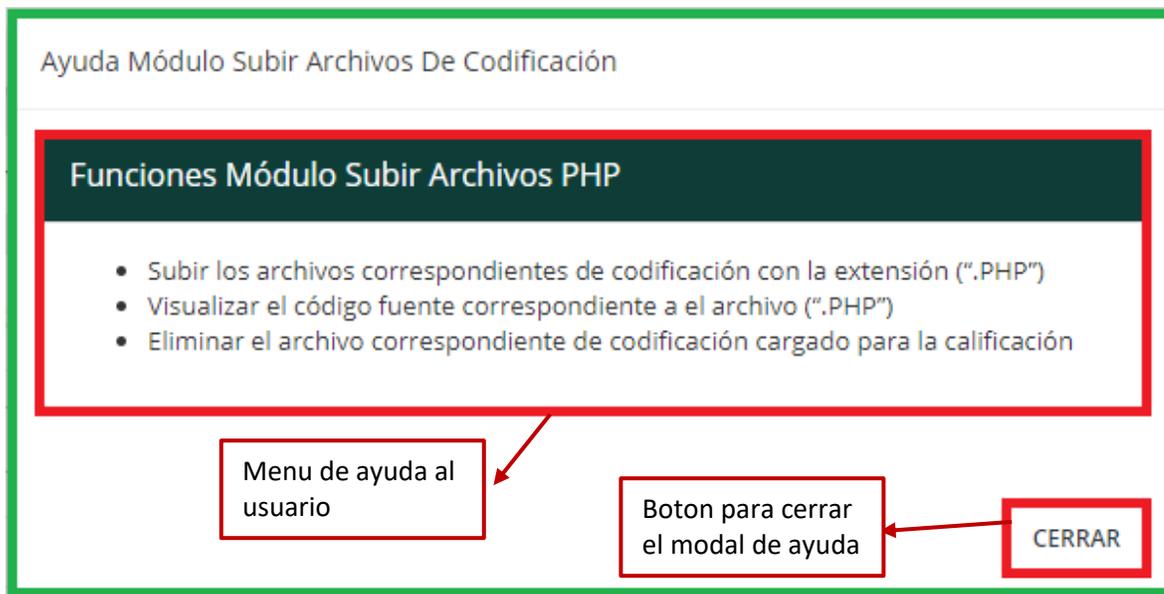


Imagen 102 Modal de ayuda para subir archivos de codificación

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

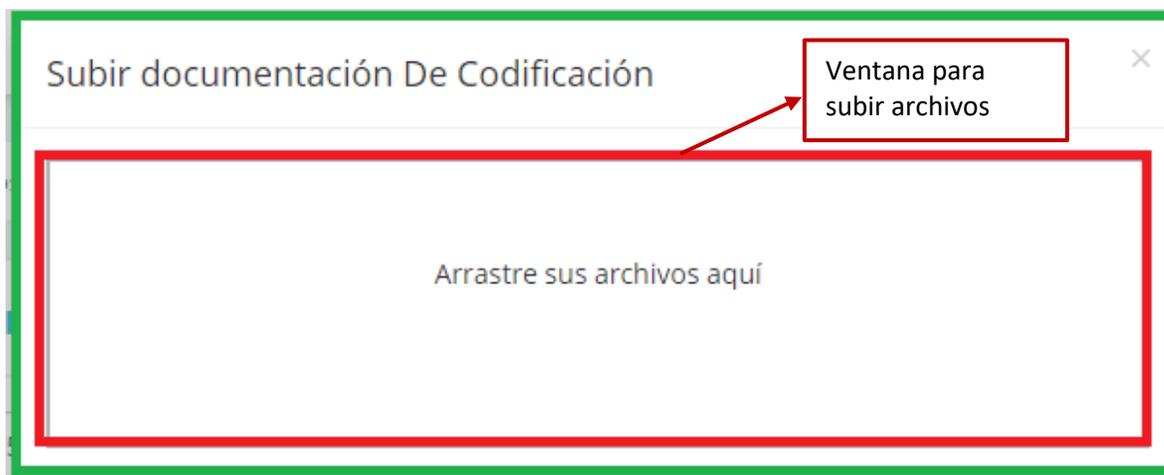


Imagen 103 Panel de carga para subir Archivos de codificación



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



Imagen 104 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



DOCUMENTOS DE CODIFICACIÓN		
Nombre Del Archivo	Estado	Operaciones
6290_UserController.php	sin calificar	 
6568_TiposDocumentoController.php		 
3567_TestInputsController.php		 
3671_TestingController.php	sin calificar	 

+ SUBIR DOCUMENTOS

Imagen 105 tabla de listado de archivos de codificación estudiante

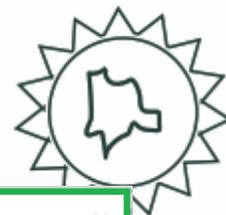
El estado del archivo cambia respectivamente luego que el profesor asignado se encargue de realizar la calificación del script, existen 2 operaciones para realizar, con el documento, la primera es eliminar el script para no ser calificado, en caso de subirlo erróneamente, se desplegara la siguiente ventana



Imagen 106 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 desplegara la siguiente vista



```
6290_UserController.php
65  /**
66   * Retorna el proyecto del usuario logeado
67   */
68  public function proyectos()
69  {
70      return request()->user()->proyectos()->with(
71          'semillero',
72          'categoria',
73          'grupoDeInvestigacion',
74          'usuarios'
75      )->get();
76  }
77
78  /**
79   * Retorna los estudiantes que no tienen
```

Visualización del código

Imagen 107 Panel de Visualización del código del desarrollador

Como tercer paso, el desarrollador debe subir el archivo SQL correspondiente a su proyecto, esto será verificado por el evaluador del proyecto, para ello debe ir a la sección de documentos y dar clic en “Base de Datos” y el sistema lo redirigirá a la siguiente vista;

DOCUMENTO DE BASE DE DATOS Modal de Ayuda

Nombre Del Archivo	Base de Datos	Operaciones
--------------------	---------------	-------------

+ SUBIR DOCUMENTO Boton para subir Archivo SQL

Imagen 84 Primera Vista del módulo de codificación

En la vista se encuentra un modal, que al presionar se desplegara un menú de ayuda y un botón de cerrar para cerrar el modal.



Ayuda Modulo Subir Archivo Sql

Funciones Modulo Subir Archivo Sql

- Subir archivo de bases de datos con extensión (".SQL"), solo permite cargar un archivo para la calificación del código.
- Visualizar el código SQL que se a cargado previamente.
- Eliminar el archivo de bases de datos cargado para la calificación.

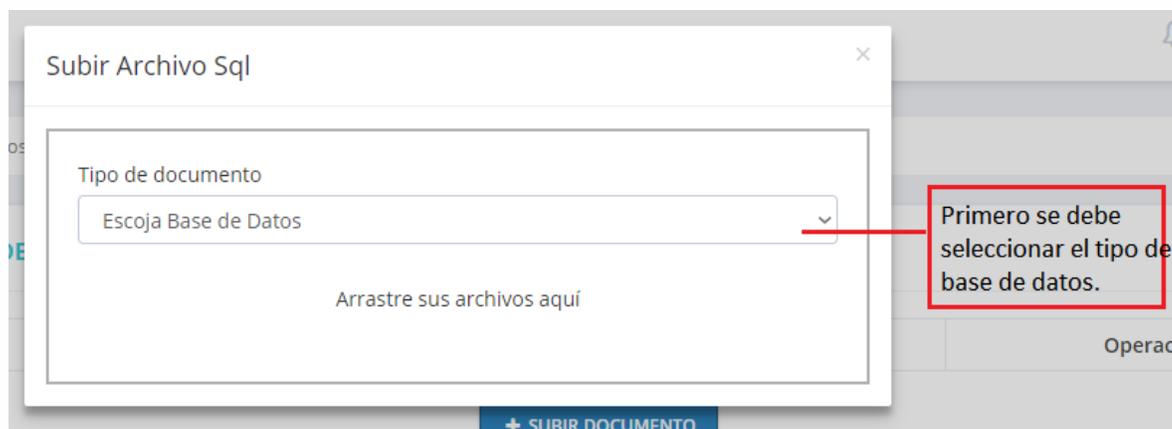
Ayuda subir
archivo SQL

Botón Cerrar
Modal

CERRAR

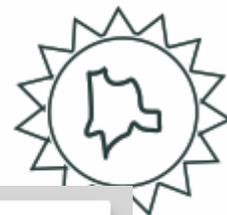
Imagen 85 Modal de ayuda para subir archivos de codificación

En la vista se encuentra un botón que permite subir el archivo SQL, y el cual genera la siguiente vista en un modal.



Teniendo en cuenta, el tipo de base de datos, y que la extensión del archivo tenga que ser ".SQL".

Sin omitir que primero se selecciona el tipo de base de datos y después si se puede arrastrar o subir el archivo.



Cuando en la vista aparezca el peso del archivo y el nombre del archivo, esto quiere decir que se subió correctamente.

DOCUMENTO DE BASE DE DATOS ?

Nombre Del Archivo	Base de Datos	Operaciones
8069_carpark.sql	PostgreSql	 
Archivo SQL subido	Base de datos seleccionada	Visualizar Archivo SQL Eliminar Archivo

Después de haber subido correctamente el archivo, aparece una vista, que contiene el nombre del archivo SQL que se subió, el tipo de base de datos que se seleccionó y unas operaciones que son, visualizar el archivo SQL o eliminar el archivo SQL.



```
8069_carpark.sql
42 -- Table structure for TBL_Carpark_Estados
43 -----
44 DROP TABLE IF EXISTS `TBL_Carpark_Estados`;
45 CREATE TABLE `TBL_Carpark_Estados` (
46   `PK_CE_IdEstados` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
47   `CE_Estados` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,
48   `created_at` timestamp(0) NULL DEFAULT NULL,
49   `updated_at` timestamp(0) NULL DEFAULT NULL,
50   PRIMARY KEY (`PK_CE_IdEstados`) USING BTREE
51 ) ENGINE = InnoDB AUTO_INCREMENT = 3 CHARACTER SET = utf8mb4 COLLATE = utf8mb4_unicode_ci ROW_FORMAT = Dyn
52 -----
53 -- Records of TBL_Carpark_Estados
54 -----
55 -----
56 INSERT INTO `TBL_Carpark_Estados` VALUES (1, 'Activo', '2018-04-11 17:40:02', '2018-04-11 17:40:02');
57 <
```

Visualización Código Fuente SQL

La Visualización del archivo le permite al alumno o desarrollador, estar seguro, si el archivo SQL que subió si fue subido correctamente a la base de datos.

De igual manera el desarrollador, podrá eliminar el archivo por si es necesario alguna corrección, teniendo en cuenta que el proyecto ya no esté en fase de calificación, por lo cual no dejara eliminar o subir un nuevo archivo.

Como cuarto paso, el desarrollador dispondrá de una interfaz gráfica, que le permitirá, diseñar y modelar la base de datos del proyecto correspondiente, esta herramienta es de uso opcional, para



ello debe dar clic a la sección de Modelación Base de Datos y el sistema lo redirigirá a la siguiente vista;

DISEÑO Y MODELACIÓN DE BASE DE DATOS



The image shows a software interface for database design. On the left side, there is a vertical toolbar with the following buttons from top to bottom: 'GUARDAR BASE DE DATOS' (green), 'OPCIONES' (red), 'AGREGAR TABLA' (blue), 'EDITAR TABLA' (blue), 'CLAVES' (blue), 'ELIMINAR TABLA' (blue), 'LIMPIAR TABLAS' (blue), 'AGREGAR CAMPO' (blue), 'EDITAR CAMPO' (blue), 'ARRIBA' (blue), 'ABAJO' (blue), 'CREAR CLAVE FORÁNEA' (blue), and 'CONECTAR CLAVE FORÁNEA' (blue). The main area is a large grid. In the bottom right corner of the grid, there is a red-outlined rectangle.

Esta vista tiene como fin ayudarle al estudiante o desarrollador en su diseño de la base de datos, teniendo en cuenta que este mismo le generara el código SQL, y si el desarrollador lo dispone, puede usar el código SQL, para hacer su calificación o evaluación en el módulo de base de datos.



The screenshot shows a database design tool interface. On the left, there is a vertical menu with the following options: GUARDAR BASE DE DATOS (green), OPCIONES (red), AGREGAR TABLA (blue), EDITAR TABLA (blue), CLAVES (blue), ELIMINAR TABLA (blue), LIMPIAR TABLAS (blue), AGREGAR CAMPO (blue), EDITAR CAMPO (blue), ARRIBA (blue), ABAJO (blue), CREAR CLAVE FORÁNEA (blue), CONECTAR CLAVE FORÁNEA (blue), and ELIMINAR CLAVE FORÁNEA (blue). In the center, a dialog box titled 'Editar tabla' is open. It contains a text input field for 'Nombre' with the value 'TBL_Usuarios' and a larger text area for 'Comentario'. At the bottom of the dialog are two buttons: 'ACEPTAR' (blue) and 'CANCELAR' (red).

La anterior, es una vista el cual permite dar a conocer que la interfaz es muy intuitiva a la hora de hacer uso de alguna herramienta, como lo es crear tablas, campos, llaves foráneas, o para generar el código SQL.

The screenshot shows the same database design tool interface. The left menu is identical. In the center, a table diagram is visible with the name 'TBL_Usuario' and a single field 'id' highlighted in yellow. In the bottom right, a dialog box titled 'Guardar Base de Datos' is open. It features a 'GENERAR SQL' button and a text area containing the following SQL code:

```
--  
--  
-- ALTER TABLE `TBL_Usuario` ENGINE=InnoDB DEFAULT CHARSET=utf8  
-- COLLATE=utf8_bin;  
--  
-- Test Data  
--  
-- INSERT INTO `TBL_Usuario` (`id`) VALUES  
-- ('');
```



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;

The screenshot displays the CALISOFT dashboard for a project named 'PROYECTO DE GRADO'. The left sidebar contains a menu with 'Evaluación' highlighted, along with other options like 'Modelación', 'Plataforma', 'Base Datos', and 'Codificación'. The main content area shows the project details, including the state 'EVALUACION', and a list of evaluators: Fabian - Fabian@email.com, johan - johan@flix.co, and Paise - paise@mail.com.

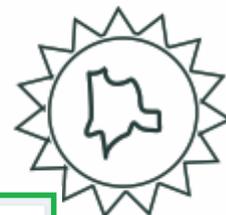
PROYECTO	
Nombre:	Proyecto de prueba
Estado:	EVALUACION
Categoría:	sistemas de información
Semillero	Linudec
Grupo de investigación:	GESAF
Creado el:	29/3/2018

INTEGRANTES	
Fabian - Fabian@email.com	
johan - johan@flix.co	

EVALUADORES	
Paise - paise@mail.com	

Imagen 108 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 diagramas, el desarrollador debe dirigirse a la sección de evaluación y dar clic en el apartado de “Modelación” y el sistema lo redirigirá a la siguiente vista;



The screenshot shows a web interface for model evaluation. At the top left, it says 'EVALUACION DE MODELADO'. The main area is divided into two columns. The left column shows a list of documents with their evaluation percentages: 'Total: 92 %', 'Casos de uso.pdf (Diagrama de casos de uso) 100%', 'Casos de uso.pdf (Diagrama de clases) 73%', 'Casos de uso.pdf (Diagrama de secuencias) 100%', 'Casos de uso.pdf (Diagrama de actividades) 93%', 'Casos de uso.pdf (Diagrama de despliegue)', and 'Casos de uso.pdf (Modelo Entidad Relación)'. The right column shows a detailed view for 'Casos de uso.pdf' with a 'Total: 100%' and a list of components: 'Actores', 'Casos de uso', 'Inclusión de casos de uso', 'Casos de uso extendidos', 'Puntos de extensión', and 'Límite del sistema', each with a checkmark. A 'Reporte' icon is in the top right. A callout box labeled 'Nombre del documento' points to the document name in the right column. Another callout box labeled 'Nombre del evaluador' points to the name 'Paisa' in the right column. A callout box labeled 'Porcentaje del total de evaluación' points to the 'Total: 92 %' in the left column. A callout box labeled 'Nombre del documento y tipo de diagrama al que pertenece' points to the first item in the left column. A callout box labeled 'Porcentaje de evaluación' points to the '93%' in the left column. A callout box labeled 'Calificación de los componentes' points to the list of components in the right column.

Imagen 109 Vista de resultados de evaluación de modelado, desarrollador

En donde puede observar la calificación que cada evaluador ha hecho sobre sus diagramas, donde puede dar clic en cada uno de los hipervínculos de la derecha y automáticamente se actualizará el apartado de la derecha, mostrando el nombre del evaluador que realizó la evaluación y los componentes que aprobó o reprobó, en la parte superior puede observar el porcentaje total que llevan los diagramas evaluados, además, si desea descargar un reporte de su evaluación, puede dar clic en el icono con forma de hoja que se encuentra en la parte superior derecha y automáticamente se descargará un reporte con el formato de la Universidad de Cundinamarca.

Para la sección de evaluación de plataforma, el desarrollador debe de realizar unos pasos, ir a la sección de evaluación y dar clic en el apartado de plataforma, a continuación, el sistema lo redireccionará a la siguiente vista;



Nombre del caso prueba

Detalles del caso prueba

Formulario en formato JSON

Botón para interpretar el texto JSON

PLATAFORMA: PROYECTO DE PRUEBA	
Propósito:	Registrar un usuario.
Alcance:	Diligenciar el formulario de registro y enviarlo.
Resultado Esperado:	Registrar un usuario con daos coherentes.
Criterios:	El nombre no puede permitir números ni símbolos, además el correo debe ser coherente y debe estar a prueba de ataques.
Prioridad:	alta
Estado:	evaluar
Plazo:	2018-03-27 00:00:00

Imagen 110 Vista caso pruebas, desarrollador

Sólo se puede acceder a este apartado cuando un evaluador crea un caso prueba, en ese momento el desarrollador es notificado y sabrá que ya puede ingresar a este módulo. En este módulo el desarrollador puede ver todos los caso prueba que se creen, por medio de los detalles podrá tener una idea más clara de los propósitos que espera el evaluador, en este ejemplo el caso prueba corresponde a un registro, entonces, el desarrollador debe ir por medio del navegador a la página donde se encuentre el formulario de registro de su proyecto, una vez estando sobre la página, se debe hacer uso de la extensión de Calisoft como se ve a continuación (Esta extensión se enseña a instalar en el manual de instalación de Calisoft). Debe dar clic derecho y seleccionar inspeccionar y en el siguiente apartado encontrará la extensión;

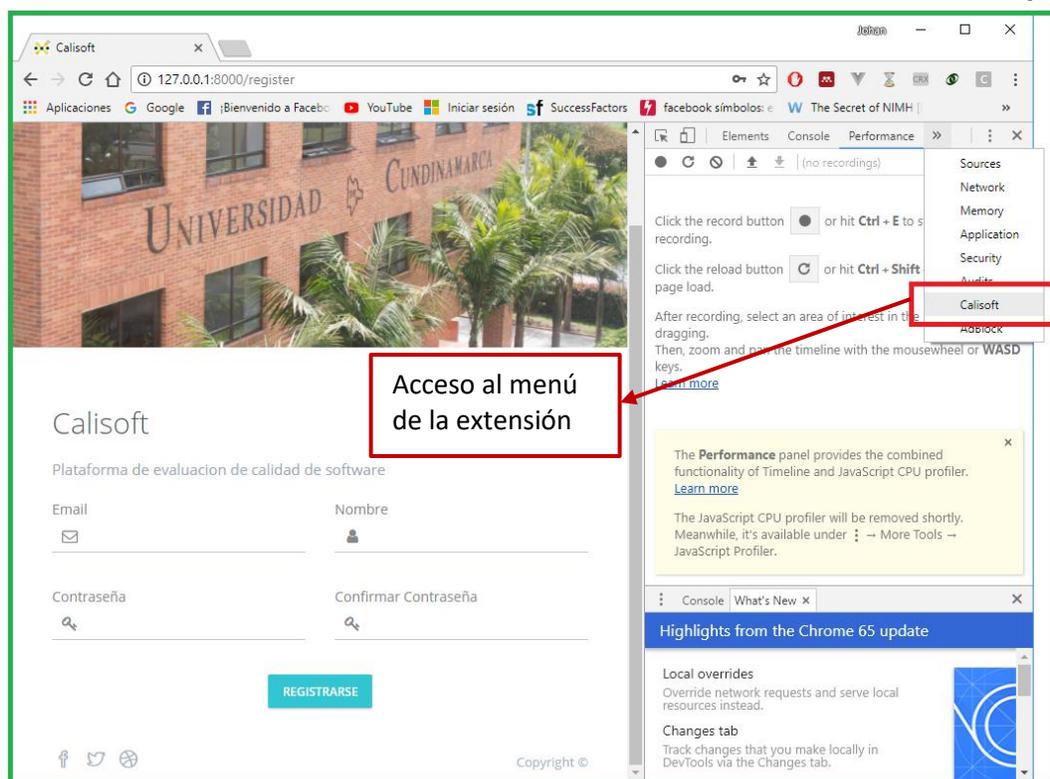


Imagen 111 Ubicación de la extensión de Calisoft

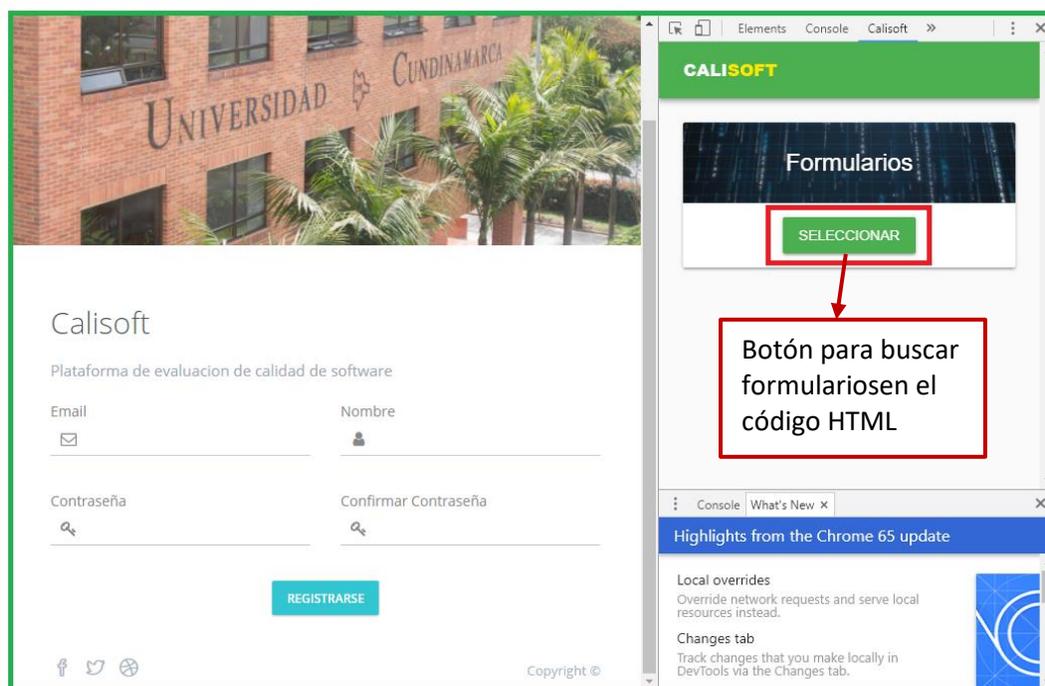


Imagen 112 Estructura de la extensión de Calisoft



Al dar clic en el botón “SELECCIONAR” automáticamente el sistema diferenciará las etiquetas que corresponden a un formulario de todo el código HTML de la página, y procederá a pintar un rectángulo sobre el formulario, el desarrollador debe dar clic sobre el rectángulo que se ubica sobre el formulario de su interés, como se ve a continuación;

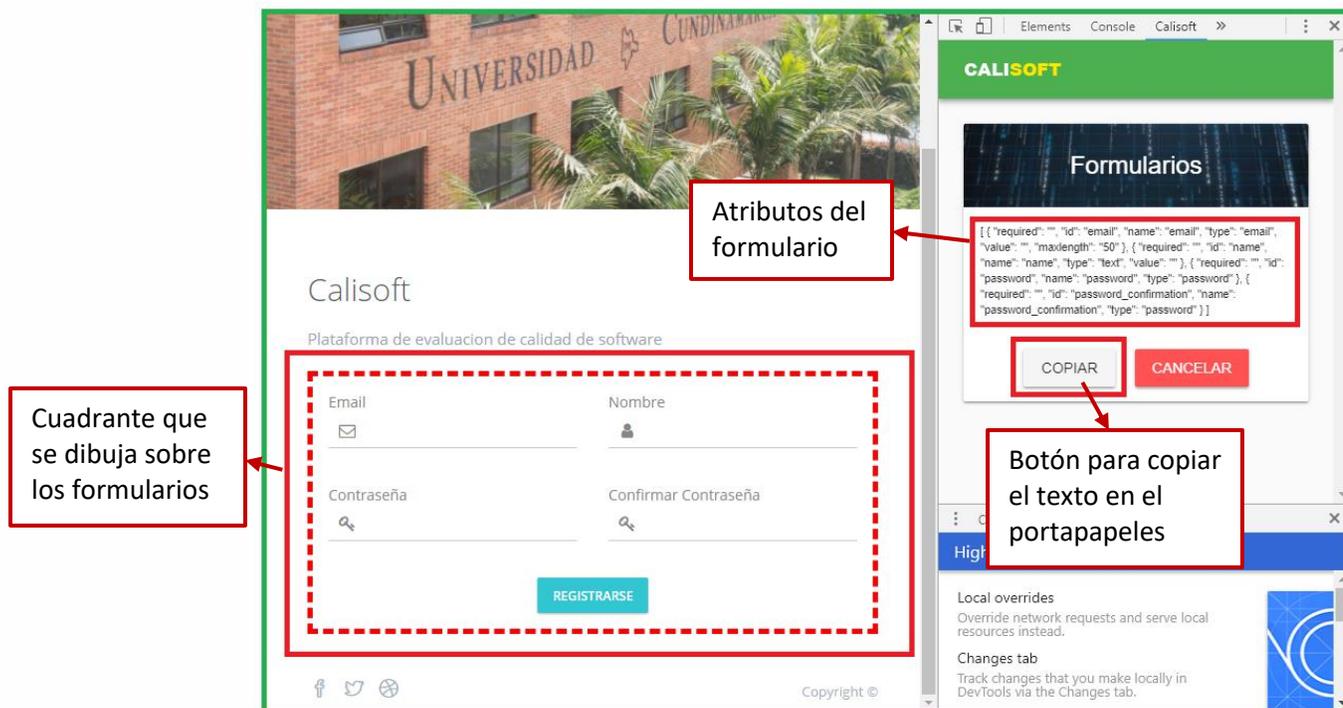


Imagen 113 Obtención de atributos de un formulario

Así se obtendrán todos los atributos del formulario y se observarán en un recuadro de la extensión, los atributos se guardan en un formato JSON, el desarrollador debe dar clic en el botón “COPIAR” y el texto se copiará en el portapapeles, ahora debe dirigirse a la vista que corresponde a la imagen 88 y pegar el texto JSON y dar clic en el botón “SIGUIENTE” y el sistema interpretará el texto y mostrará el siguiente modal;



The image shows a modal window titled "Enviar Caso Prueba" with a close button (X) in the top right corner. The form contains the following elements:

- An "Observación (Opcional)" text area at the top.
- A vertical list of four dropdown menus on the left side.
- A table on the right side listing input attributes:

NOMBRE: EMAIL	ID:EMAIL
NOMBRE: NAME	ID:NAME
NOMBRE: PASSWORD	ID:PASSWORD
NOMBRE: PASSWORD_CONFIRMATION	ID:PA

At the bottom of the form is a green button labeled "SUBIR CASO PRUEBA".

Annotations with red boxes and arrows point to the following elements:

- "Observación sobre el formulario enviado" points to the text area.
- "Escoger si el input pertenece a el nombre de una persona" points to the dropdown menus.
- "Botón para subir caso prueba" points to the "SUBIR CASO PRUEBA" button.
- "Atributos name y id de los inputs encontrados" points to the table of attributes.

Imagen 114 Modal para enviar caso prueba. desarrollador

Donde el desarrollador puede adjuntar una observación que desea que vea el evaluador, en la zona de abajo se encuentra una serie de opciones para escoger qué input evalúa el nombre de una persona, a la parte derecha se ilustran los nombres e ID's de los inputs encontrados en el código HTML con el objetivo de que del desarrollador pueda diferenciarlos fácilmente, una vez realizadas las opciones anteriores se debe proceder a dar clic en el botón "SUBIR CASO PRUEBA" y el sistema notificará al evaluador para que realice el proceso de Testing sobre el formulario. El sistema notificará al desarrollador cuando sus casos de prueba pasen por el testeo. Cuando eso suceda, el caso prueba en la vista de plataforma se actualizará de la siguiente forma;



PLATAFORMA: PROYECTO DE PRUEBA

Ver reporte es formato pdf.

PRUEBAS DETALLES

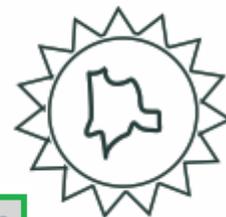
Propósito:	Registrar un usuario.
Alcance:	Diligenciar el formulario de registro y enviarlo.
Resultado Esperado:	Registrar un usuario con daos coherentes.
Criterios:	El nombre no puede permitir números ni símbolos, además el correo debe ser coherente y debe estar a prueba de ataques.
Prioridad:	alta
Estado:	evaluar
Plazo:	2018-03-30 21:43:01
Json Enviado:	Total de Inputs: 4
Calificación:	50%

Detalles de las pruebas realizadas

Calificación obtenida

Imagen 115 Resultados de la calificación de los casos de prueba, desarrollador

El desarrollador puede observar en la parte inferior la calificación que obtuvo, se habilitará un botón "PRUEBAS" que al dar clic se despliega el siguiente modal;



Pruebas: Registro			
Prueba #1		Calificación	100%
Nombre	Entrada	Estado	
email	oserna@yahoo.com	✓	
name	dolore	✓	
password	-3(eom8_w9p	✓	
password_confirmation	+wc@ n	✓	
Prueba #2		Calificación	25%
Nombre	Entrada	Estado	
email	0 OR 1=1	✓	
name	" or ""="	✗	
password	"x"; DROP TABLE members; --"	✗	
password_confirmation	"x"; DROP TABLE members; --"	✗	
Prueba #3		Calificación	25%
Nombre	Entrada	Estado	

Imagen 116 Modal detalles de la evaluación del caso prueba

En donde el desarrollador puede ver con detalle los parámetros que uso el sistema sobre sus inputs y así ver en que fallo en el proceso de evaluación, si el evaluador desea, puede volver a activar el proyecto para que el desarrollador realice las correspondientes correcciones. Si el desarrollador desea un reporte para agendar a su proyecto, puede dar clic en el icono de "hoja2 que se encuentra en la parte superior y podrá descargar el reporte en formato pdf. La estructura del reporte es la siguiente;



Información de la Universidad de Cundinamarca



UDECA
UNIVERSIDAD DE CUNDINAMARCA

CALISOFT

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

Proyecto de prueba - Evaluación de plataforma

Detalles del caso prueba

Registro	
Propósito:	Registrar un usuario.
Alcance:	Diligenciar el formulario de registro y enviarlo.
Resultado Esperado:	Registrar un usuario con datos coherentes.
Criterios:	El nombre no puede permitir números ni símbolos, además el correo debe ser coherente y debe estar a prueba de ataques.
Prioridad:	alta

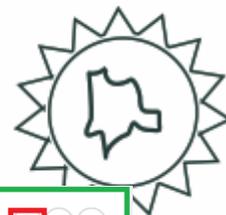
Detalles de los parámetros que se usaron en cada input

Calificación:		50%
Prueba #1	Calificación	100%
Nombre	Entrada	Estado
email	osema@yahoo.com	Correcto
name	osore	Correcto

Imagen 117 Reporte de calificación del caso prueba

El reporte dispondrá de los datos generales de la Universidad de Cundinamarca, acompañado de los detalles del caso prueba y debajo se encuentra de manera detallada los parámetros que se usaron en el proceso de calificación en cada uno de los inputs que contenía el formulario tipo JSON que subió el desarrollador.

Para ver 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



<> EVALUACION DE CODIFICACIÓN

Mostrar 5

Reporte final de todos los Scripts

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

Boton Modal de los items evaluados

Imagen 118 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 119 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

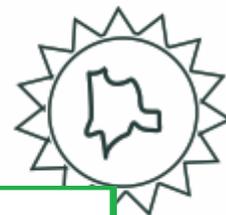


UserController.php			
Item	Atributo	Fila	Aprobado
Variables	users	16	✓
Variables	request	30	✓
Variables	user	44	✓
Variables	usuario	60	✓
Clases	UserController	14	✓
Funciones	index	30	✓
Funciones	store	41	✓
Funciones	destroy	60	✓
Funciones	proyectos	68	✓
Funciones	searchFreeStudents	82	✓
Funciones	searchEvaluators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

Imagen 120 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. Facativá
(+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 121 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)

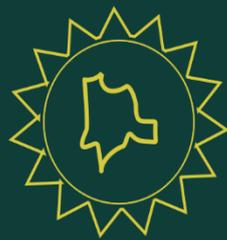
71.5%

Imagen 122 Nota final calculando el índice de cohesión o aceptación Desarrollador



5. CONTROL DE CAMBIOS DEL MANUAL

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



01-CER35041

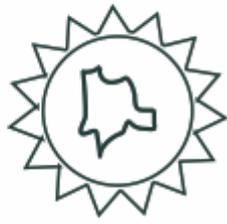


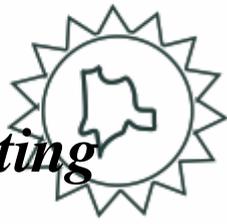
00-SC-CER35037



00-CER35037

3.1.2.ARTÍCULO 1





CALISOFT platform reengineering implementing methods of no heuristic testing for quality evaluation to software products

Cesar Yesid Barahona Rodríguez
Universidad de Cundinamarca
Facatativá, Cundinamarca, Colombia
cbarahona@ucundinamarca.edu.co

Stevenson Marquez Rangel
Johan Suarez Campos
Universidad de Cundinamarca
Facatativá, Cundinamarca, Colombia
stevensonxmarquez@gmail.com
johancamilosuarezcampos@gmail.com

Abstract— This article exposes how using non-heuristic Testing methods for the qualification of software products in the University of Cundinamarca; The non-heuristic methods will evaluate the usability of the software, in order that the final product to be evaluated is more comprehensive and understandable for the different users. These methods will be integrated into CALISOFT (Software Quality), a functional platform of the University of Cundinamarca aimed at evaluating the quality of software products.

Non-heuristic methods will be implemented to the different CALISOFT modules, with the aim of making more accurate, faster and more efficient assessment. The non-heuristic methods will be programmed, which will cause the products to be evaluated under the same standard. A mixed investigation method will be used to evaluate the results obtained and thus determine the importance and the change that the implementation of these methods give.

Keywords— Automation; usability engineering; validation; quality; Testing; module integration; non-heuristic methods.

I. INTRODUCTION

El ser humano siempre ha querido optimizar su trabajo, lo que llevo a la creación de las maquinas, y con el pasar del tiempo el hombre ha ido mejorándolas para que su usabilidad sea más óptima y sencilla. La usabilidad se puede definir como la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso. [1]

Se empleará la usabilidad para mejorar la experiencia del usuario, la ingeniería de la usabilidad posee diferentes formas para evaluar los productos Software, una de ellas es el test con usuarios (Testing) que consiste en darles instrucciones a los usuarios y a medida que pruebe el producto software, irá registrando los problemas que pueda presentar. [2]

Existen dos métodos de testeo, heurístico y no heurístico. El método heurístico es realizado por personas expertas en usabilidad y con conocimientos avanzados sobre el propósito

del software a tratar. El método no heurístico es el que se empleará en este proyecto, que consiste en pruebas realizadas por usuarios de conocimientos comunes. [3]

Actualmente en la Universidad de Cundinamarca existe una plataforma funcional llamada CALISOFT, la cual evalúa la calidad de productos software de forma satisfactoria, pero esta labor tarda alrededor de tres semanas, llegando a un punto a ser agotadora por parte del evaluador, ya que debe evaluar todos los formularios bajo su propio concepto y la cantidad de campos de entrada es elevada en cada formulario. Se programarán los métodos no heurísticos para que la evaluación de los productos software sea menos dependiente del evaluador, de esta forma se certificará una misma rigurosidad y calidad en todos los productos software que se vayan a evaluar, el evaluador tendrá que seguir unas instrucciones básicas y mínimas. [4] De esta forma se espera reducir visiblemente el tiempo en que se están tardando el testeo normal, además, para aumentar la calidad de evaluación, se integrarán dos módulos externos destinados a evaluar los estándares de codificación y nomenclatura de base de datos del producto software a tratar, de esta forma la evaluación será más integra, ya que el producto software se dividirá y se podrá evaluar cada parte de manera más detallada.

II. IMPLEMENTING NO HEURISTIC METHODS

Los métodos no heurísticos serán implementados en el módulo de pruebas funcionales de CALISOFT, se creará una base de datos destinada a almacenar la posible información que exigirán los diferentes formularios de los productos software a evaluar, se complementará con una simulación del formulario al que se le realizarán las pruebas funcionales, los campos de entrada del mismo serán llenados según su tipo con la información proveniente de la base de datos anterior mencionada.

La información almacenada en la base de datos que se utilizará para llenar los formularios estará dividida en tres



partes, información válida y segura para el campo de entrada del formulario, información no válida para el campo de entrada e información que puede ser potencialmente peligrosa e insegura para el funcionamiento del aplicativo web en proceso de evaluación.

El proceso de simulación irá integrado en componentes web, que permitirán modular los elementos usados en CALISOFT, con el objetivo de usar pequeños elementos que sean independientes unos de los otros, de esta forma crearemos elementos más complejos y eficientes. [5] Según la respuesta de la ejecución del formulario, se entregará un reporte con base al resultado de la prueba en función de lo que debió responder en su momento el aplicativo con respecto a la entrada que proporcione el sistema.

III. EXTERNAL MODULES INTEGRATION.

Las integraciones de los módulos externos harán de CALISFOT una evaluación de calidad más integra. Uno de los módulos externos está destinado a evaluar el estándar de codificación, esto consiste en evaluar las normas y reglas del lenguaje de programación por medio de la práctica de matrices enfocadas a una clasificación. [6]

De esta forma se espera promover los conceptos de confiabilidad de código, mejorando el proceso de mantenimiento y portabilidad, haciendo que el código pueda ser mejor comprendido por personas que tengan la autorización de acceder a esta parte del producto software. De esta manera se certificará la integridad del producto y a la vez su calidad.

El segundo módulo tiene como objetivo valorar la nomenclatura y modelación de la base de datos que se le exija al producto software a evaluar, aumentando la certificación de calidad. Éste módulo trabajará bajo la norma de estándares de calidad ‘ISO/IEC 25012- ISO/IEC 19139’. [7]

La norma ‘ISO/IEC 25012- ISO/IEC 19139’ aporta varias ventajas, como son; “Asegurar la legibilidad del modelo de datos, inclusive para personas que no están relacionadas con el ambiente informático, en etapas de análisis y diseño, y facilitar la tarea de los desarrolladores, programadores y estudiantes en el desarrollo de software”. [8] De esta manera se asegurará la legitimidad del modelo de datos, exigiendo el estándar a los productos software a evaluar.

IV. VISUAL IMPROVEMENT.

Idealmente la usabilidad debe considerarse desde el inicio del proceso de desarrollo hasta el final del sistema, Antes de iniciar un proyecto, es recomendable tener una idea acerca de las características de la usabilidad para implementarlas en

nuestro producto y suplir las necesidades del usuario. Una de las características principales de la usabilidad es la facilidad de aprendizaje, que es: “La facilidad de aprender la funcionalidad y comportamiento del sistema. Define en cuánto tiempo un usuario, que nunca ha visto una interfaz, puede aprender a usarla bien y realizar operaciones básicas. Debido a que la Interfaz de Usuario (IU) es la puerta hacia la funcionalidad del sistema subyacente, es necesario diseñarla de tal forma que sean usables para los usuarios, ya que un mal diseño se convierte en un factor que limita su uso.” [9]

Por tal razón se mejorará la interfaz gráfica de CALISOFT, con la finalidad de que sea más óptima y se acople de manera eficaz a las necesidades del usuario final. Ésta mejora en el diseño se trabajará bajo el estándar establecido por el centro de innovación y tecnología de la universidad de Cundinamarca (CIT). Además, se simplificarán las transacciones de recursos en la plataforma de forma que se establezca una sola vista para las diferentes operaciones que gestiona el aplicativo. El nuevo diseño de la plataforma hará uso del dinamismo y la creatividad frente a los cambios en la información, para presentarle al usuario el resultado de la transacción de la forma más rápida y clara. [10]

V. USABILITY METRICS.

Las métricas evalúan los atributos o características de la usabilidad que puedan ser medibles. Beltré Ferreras [11] define los siguientes atributos: la facilidad de aprendizaje, el recuerdo en el tiempo, la eficiencia en uso, la tasa de errores y la satisfacción.

RELACION DE ATRIBUTOS MEDIBLES DE LA USABILIDAD

<i>Atributo</i>	<i>Significado</i>	<i>Forma de medir</i>
Facilidad de aprendizaje	Implica cuán rápido y fácilmente los usuarios pueden comenzar a realizar un trabajo productivo con un sistema que usan por primera vez.	Tiempo que el usuario final utiliza el sistema antes de alcanzar el nivel de eficiencia que tiene el usuario experto en el uso del aplicativo.
Recuerdo en el tiempo	Capacidad del sistema de permitir al usuario utilizar la aplicación siempre, sin tener que recordar su funcionamiento.	Tiempo requerido para concluir la actividad.
Eficiencia en uso	Productividad del usuario con el uso del sistema.	Número de tareas por unidad de tiempo en que el usuario (experto) es capaz de utilizar el sistema.

Tasa de errores	Errores cometidos durante el uso del sistema y cuán fácil el usuario se recupera de ellos, tanto del número como del tipo de errores	Número de errores que el usuario comete cuando intenta realizar una tarea concreta y como se recupera del error.
Satisfacción	La opinión subjetiva que se forma el usuario acerca del sistema.	Cuestionarios de satisfacción que llena los usuarios.

VI. AUTOMATION PROCESS

“La calidad del producto depende en gran medida de la calidad del proceso” [12] Dicho lo anterior, para garantizar una calidad en el software, hay que tener en cuenta la rigurosidad de las pruebas que se realicen antes de dar por aceptado el aplicativo, por esta razón, el testeo no heurístico se programó para que su ejecución sea automática y así, evaluar todos los aplicativos bajo un mismo estándar, estándar que certifica su calidad. Los pasos del proceso son los siguientes:

- En el aplicativo el evaluador podrá ver con facilidad los requerimientos del software a evaluar, teniendo en claro los requerimientos, el evaluador procederá a crear los casos de uso correspondientes a cada requerimiento.
- Al crear el caso de uso, se le notificará al desarrollador del software y este deberá dirigirse a la vista en dónde se encuentra el formulario correspondiente a dicho caso de uso. El desarrollador extraerá una copia de los atributos del input por medio de una extensión del navegador de Chrome, llamada CaliExtens, CaliExtens fue creada para obtener los atributos Html de los inputs que se alojen en cualquier formulario.

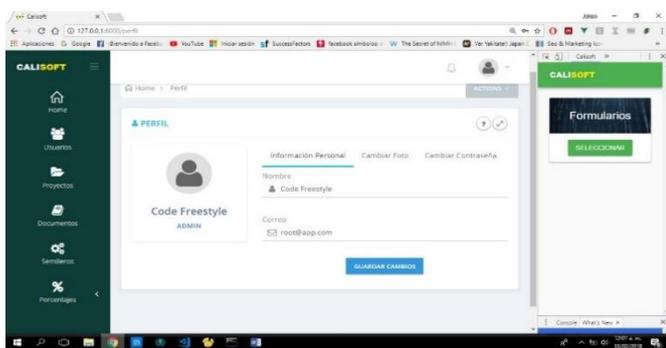


Fig 1. Extension de CALISOFT

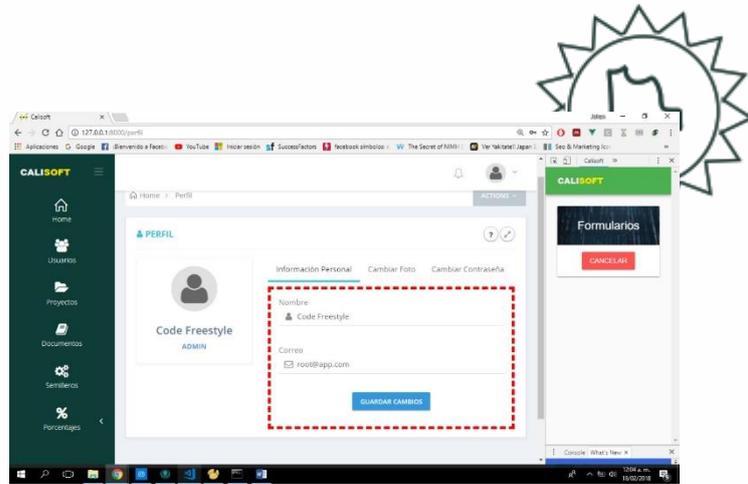


Fig 2. Selección de formulario

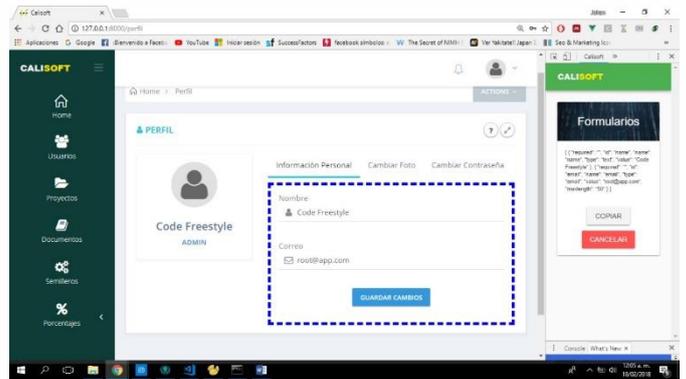


Fig 3. Extracción de formulario

- Al tener la copia de los atributos, el desarrollador deberá seleccionar el caso de uso correspondiente e ingresarlos al sistema.
- Tras el paso anterior, se le notificará al evaluador de la actualización del caso de uso y este procederá a ejecutar el testeo automatizado.
- En este paso, Calisoft procederá a emular el formulario que el desarrollador envió (esto se logra gracias a los atributos obtenidos por CaliExtens), y el sistema procederá a comunicarse con la base de datos y por medio de iteraciones, irá ingresando los correspondientes valores para cada tipo de input, como si un usuario experto e inexperto lo diligenciara, en tiempo real, un script evaluará el contenido, y notificará y almacenará los resultados positivos o negativos en cada iteración.
- Al finalizar el proceso, el evaluador observará el reporte, sí el resultado es negativo, el evaluador le notificará al desarrollador su falencia y volverá a habilitar el caso de uso para que el desarrollador envíe de nuevo los atributos con sus correspondientes correcciones, este proceso repetirá hasta que no presente errores en el proceso de testeo.



Una vez obtenido un valor positivo en el reporte, se procede a cerrar el caso de prueba y el resultado es guardado para un reporte final.

VII. METODOLOGY.

Se implementó una metodología de investigación mixta para evaluar la usabilidad del producto. Esta metodología se caracteriza por integrar tanto el método cualitativo como el método cuantitativo usando las fortalezas de cada método con el fin de obtener un panorama más completo del fenómeno.

¿Las técnicas no heurísticas programadas mejorarán la evaluación de calidad de los productos software a calificar?

Para comprobar que la implementación de técnicas no heurísticas programadas mejora la evaluación de calidad, se realizó una evaluación conforme a un criterio cuantitativo, en donde se realizaron evaluaciones no heurísticas manuales y evaluaciones no heurísticas programadas en un periodo de una semana a un mismo aplicativo web. Los datos por obtenidos fueron; cantidad de módulos evaluados, cantidad de páginas evaluadas en cada módulo, cantidad de datos ingresados en cada módulo, cantidad de errores presentados en cada módulo. Este proceso se repitió por dos semanas más y con un aplicativo web diferente por cada semana. Con los datos obtenidos se hizo una comparación entre la labor del evaluador humano y el evaluador programado, donde se observó en los resultados una gran diferencia entre las cantidades de los datos de ambas evaluaciones. Con los resultados obtenidos se pudo analizar y definir que el método programado fue más eficaz para evaluar la calidad de un producto software.

¿La integración de módulos externos mencionados anteriormente hará que el proceso de calificación sea más íntegro y eficaz?

Para comprobar que la integración de módulos externos definirá con mayor exactitud la calidad de un producto software, se realizó una evaluación conforme a un método mixto, en donde se investigó y analizaron los factores principales que definen la calidad de un producto software, buscando a fondo en cada característica y se comparó con las nuevas características que aportaron los módulos a integrados mencionados en este documento. Además, Al realizar el estudio del estado de arte, se encontraron estudios de calidad que se pueden implementar en CALISOFT para que sea más íntegro y eficaz a la hora de definir la calidad de un producto software, dichos estudios no se implementaron, pero se dejan en el apartado de conclusiones para una futura implementación. Al analizar esta evaluación se demostró que los módulos integrados hacen de CALISOFT un software más íntegro y eficaz a la hora de definir la calidad de un producto software.

VIII. CONCLUSIONES.

De acuerdo con lo expuesto en el presente documento, la implementación de métodos no heurísticos hizo que la evaluación de calidad fuera más íntegra, confiable y eficaz, porque se implementaron métodos programados, y la evaluación se realizó bajo un mismo estándar a los diferentes productos software evaluados, obteniendo menor tiempo de resultados y una facilidad de uso notable. Se espera implementar en un futuro, un módulo que evalúe la calidad visual del software en diferentes plataformas (computadores, tablets, celulares, etc.) realizando una comparación de imágenes mediante el procesamiento digital de la pantalla.

IX. REFERENCES

- [1] M. Mascheroni, C. Greiner, and R. Petris, "Calidad de software e Ingeniería de Usabilidad," in *WICC 2012*, 2012, no. 11112, pp. 656-659.
- [2] M. Bolaños-Pizarro and R. Vidal-Infer, A; Navarro-Molina, Carolina; Valderrama-Zuriani, Juan Carlos; Aleixandre-Benavent, "Usabilidad: concepto y aplicaciones en las páginas web médicas," *Papeles Médicos*, vol. Vol 16, Nr, no. 1, pp. 14-21, 2007.
- [3] M. A. Mascheroni, C. L. Greiner, G. N. Dapozo, and M. G. Estayno, "Ingeniería de Usabilidad. Una Propuesta Tecnológica para Contribuir a la Evaluación de la Usabilidad del Software," *Rev. Latinoam. Ing. Softw.*, vol. 1, no. 4, p. 125, May 2014.
- [4] I. Acosta, E. Nieto, and C. Barahona, "Metodología para la evaluación de calidad de los productos software de la Universidad de Cundinamarca," vol. 3, no. 2.
- [5] J. F. S. Carmona, "Diseño e implementación de un sistema de testing automatizado y con validación de web components integrados en aplicaciones web," *Univ. Politécnica Madrid Esc. Técnica Super. Ing. Informáticos*, p. 112, 2016.
- [6] D. V. Mackellar, "Injection of business coding standards practices to embedded software courses," *Proc. - Front. Educ. Conf. FIE*, vol. 2016-Novem, 2016.
- [7] Power Data, "La Calidad de Datos, los factores imprescindibles a tener en cuenta en una corporación.," 2012.
- [8] R. Por, G. Ignacio, and Q. Alvarado, "Estándares de Bases de Datos," 2009.
- [9] W. O. Sánchez and W. Ovidio, "La usabilidad en Ingeniería de Software: definición y características," 2013.
- [10] C. Y. B. Rodríguez, D. S. A. Rojas, and P. A. C. Rodríguez, "Técnica híbrida de estimación basada en el análisis de puntos de función y puntos de casos de uso," *Memorias Congr. UTP*, vol. 0, no. 0, pp. 102-109, 2017.
- [11] Hayser Jacquelin Beltré Ferreras., "Aplicación de la usabilidad al proceso de desarrollo de páginas web," *Dr. Diss. Inform.*, 2008.
- [12] M. L. Universidad Pedagógica y Tecnológica de Colombia. Facultad de Ingeniería. Centro de Estudios y Educación Continua., F. J. Pino-Correa, and J. M. Martínez, *Revista FI-UPTC: publicación del Centro de Estudios y Educación Continua de la Facultad de Ingeniería, CEDEC.*, vol. 24, no. 39. Universidad Pedagógica y Tecnológica de Colombia, 2015.

3.1.3.ARTÍCULO 2

Pruebas funcionales automatizadas para evaluar la calidad en productos software, implementando métodos de Testing no heurístico

Cesar Yesid Barahona Rodríguez
(Author)

Docente Universidad de
Cundinamarca
Bogotá, Colombia
cbarahona@ucundinamarca.edu.co

Stevenson Marquez Rangel
(Author)

Estudiante Universidad de
Cundinamarca
Bogotá, Colombia
stevenson@openmailbox.org

Johan Suarez Campos
(Author)

Estudiante Universidad de
Cundinamarca
Bogotá, Colombia
stevenson@openmailbox.org

Abstract— Este artículo expone como se implementaron métodos automatizados de Testing no heurísticos en la plataforma CALISOFT (Calidad de Software) de la Universidad de Cundinamarca. Esta plataforma contiene diferentes módulos que evalúan la estructura de un producto software para así certificar su calidad frente al mercado, actualmente la plataforma cuenta con un módulo que realiza pruebas funcionales de forma automática gracias a la implementación de dos componentes que se explicarán en este artículo, este proceso de automatización se implementa con el objetivo de realizar evaluaciones más precisas, rápidas y con un alto grado de eficacia.

Keywords— *Pruebas funcionales, Testing, métodos no heurísticos, validación, automatización.*

I. INTRODUCCIÓN

Este artículo se basa en explicar la implementación y el funcionamiento de pruebas funcionales automatizadas implementadas en procesos de evaluación de calidad sobre productos software, con el objetivo de aumentar la calidad y disminuir tiempos de ejecución. Se implementaron una serie de pruebas funcionales automatizadas sobre un módulo de Testing proveniente del software CALISOFT (Calidad de software) desarrollador en la Universidad de Cundinamarca, el proceso consiste en

implementar métodos de Testing no heurístico, que es un método de evaluación de calidad proveniente de la ingeniería de la usabilidad [1].

II. ANTECEDENTES.

A la fecha se considera cualquier actividad tecnológica como un factor crítico en la vida del ser humano, dado que la tecnología ha avanzado tan rápidamente que muchas actividades cotidianas han pasado a ser suplementadas por software que reducen tiempo y costos, por ejemplo, en transacciones electrónicas, telemedicina, negocios empresariales y entre otras más. Un error en el software puede generar grandes complicaciones en la vida del ser humano [2].

Los siguientes son ejemplos de mal funcionamiento de un software en condiciones no favorables:

- El 4 de junio de 1996, ocurrió un error en el primer vuelo del cohete Ariane 5, el cohete perdió amplitud y orientación causado por el fallo del motor pocos segundos después del lanzamiento,

generando una explosión y pérdida de información. Esto se debió al hecho de que el código con el que se programó el despegue, fue reutilizado de su antecesor, el Ariane 4, el código no era compatible con el nuevo modelo y al no realizar pruebas de software, el error ocasionó la destrucción de la base del lanzamiento. [3]

- Entre los años 1985 a 1987 en Canadá, ocurrieron una de accidentes a causa de la máquina Therac-25, que empleaba radiación para realizar terapias, los incidentes dejaron a varias personas muertas y a otros severamente lesionado. La raíz del problema, un error en la programación de la interfaz gráfica, el software permitía trabajar con altas radiaciones que el cuerpo humano no puede soportar.[4] El software antes de producción debía ser evaluado con múltiples entradas de datos, lo cual no se realizó cayó en demandas.

III. IMPORTANCIA DE LAS PRUEBAS FUNCIONALES

La seguridad es el factor principal, dentro del modelo de requerimientos se encuentra como un atributo no funcional directamente responsable de la calidad del software, conservando y previniendo la pérdida de información conservando la integridad de los datos [5].

Basados en la norma ISO 9000, la definición de calidad es entendida como un conjunto de propiedades o características inherentes de un producto, que cumplen con una serie de requisitos planteados y que logran satisfacer las necesidades de un usuario [5].

Para mitigar los riesgos y disminuir perdidas tanto económicas y de clientes, la industria del software le da importancia al proceso de pruebas funcionales que

certifiquen la calidad del producto final, fortaleciendo la entidad y aumentando su competitividad [6].

IV. PRUEBAS DE SOFTWARE

Las pruebas son las encargadas de verificar y controlar el buen funcionamiento del software, para ello se debe llevar un orden correcto para alcanzar los objetivos propuestos y así cumplir con los requerimientos del sistema, se debe empezar por la creación de casos de prueba, cuya finalidad es observar y prever como será el funcionamiento y las respuestas adecuadas que debe dar el sistema frente a múltiples acciones. Los resultados determinan si el sistema cuenta con errores en su programación, por ende, en el código del software. Hay estudios donde los casos de prueba son obtenidos mediante los modelos de casos de uso que se crean al inicio de todo proyecto, además de usar otro componente en la arquitectura que son los requerimientos del sistema, y también aconsejan realizar un cruce entre ambos para obtener una mejor definición de los casos de prueba. [7].

Cabe resaltar que, en este proyecto de automatización de pruebas funcionales, los casos de prueba que se realizan en el sistema son obtenidos mediante el estudio de los modelos de casos de uso más los requerimientos del proyecto a evaluar.

El principal desafío es evaluar un software en el menor tiempo posible, con el objetivo de mantener la calidad del producto software siempre en un buen estándar y entregarlo en un ciclo corto, como solución a esta problemática se utilizan pruebas automatizadas que reduzcan los tiempos de entrega y que optimice la calidad del producto a evaluar. [8]

V. PLANTAMIENTO DEL PROBLEMA

Actualmente en la Universidad de Cundinamarca existe un software llamado CALISOFT, este software tiene como propósito evaluar la calidad de un producto

software con base a su documentación, código base del software, nomenclatura de la base de datos y un módulo que evalúa los parámetros de entrada que puedan recibir los múltiples formularios del software a evaluar. Anteriormente durante el proceso de evaluación, el evaluador tardaba entre dos o tres días para realizar correctamente las pruebas de calidad sobre los formularios, cada parámetro de entrada usado era dado por la creatividad o experiencia del evaluador, esto ocasionaba que no todos los productos software se evaluaran de la misma manera, dado que cada evaluador tenía su propia manera de evaluar, además, habían casos en los que no se probaban de todas las formas posibles o necesarias los diferentes campos de entrada de los formularios.

Como solución a esta problemática se hace una implementación de métodos de Testing no heurísticos programados en el módulo de Testing, con la finalidad de reducir tiempos de ejecución y que el evaluador ya no tenga que disponer de sus conocimientos para realizar pruebas dado que los parámetros son programados y la evaluación se hace de forma automática.

VI. IMPLEMENTACIÓN

Para su implementación se crearon dos tablas en la base de datos, que son las siguientes;

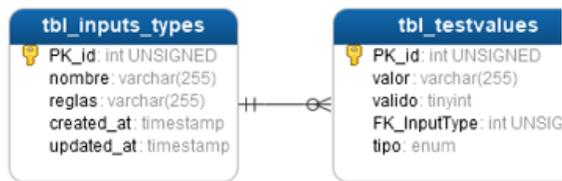


Fig. 1 Tablas destinadas a almacenar la información que se usará en el proceso de Testing automatizado.

En la tabla “Tbl_inputs_types” se almacenan los diferentes tipos de inputs que puede haber en un formulario, por ejemplo, un tipo email, contraseña, text, number, etc. Entre sus campos se puede guardar el nombre del tipo, las reglas con las que se valida la información

y los datos que ingresan mediante ese tipo de inputs y dos campos que almacenan la fecha en que se crea y la fecha en que se realice una modificación desde el sistema. Y la tabla “Tbl_testvalues” almacena los diferentes valores con los que trabajará el proceso de Testing no heurístico, en el campo de valor se asignará una sentencia que corresponda al tipo de input en el que se utilizará, por ejemplo, para el tipo email, en valor se almacenará una sentencia como; prueba@email.com o pruebaemail.com, como se puede observar, una sentencia es correcta y la otra es incorrecta, ese valor de incorrecto e incorrecto se almacena en el campo de la tabla “valido”, donde 1 será correcto y 0 incorrecto. La columna FK_InputType corresponde a la llave foránea de la tabla Tbl:inputs:types, y el campo llamado “tipo” corresponde a al tipo de ataque, si es html, xss o sql.

El proceso cuenta con la intervención de dos actores, un evaluador y un desarrollador. Como paso inicial, el desarrollador debe subir toda la documentación de su proyecto, dentro de la documentación se encontrarán los diagramas de casos de uso, con base a los diagramas de casos de uso, el evaluador procede a crear casos de prueba donde vea que hay un ingreso de datos al software del desarrollador.

Fig. 2 Formulario de creación de un caso prueba.

Con los casos de prueba creados, el desarrollador tiene como objetivo dirigirse al formulario que corresponda y crear una copia del formulario por medio de la extensión de Calisoft.



Fig. 3 Interfaz d de Calisoft que ayuda en la obtención de atributos de los inputs encontrados en un formulario en código Html.

La extensión se sitúa en el apartado del inspector de páginas, el cual se accede oprimiendo clic en la ventana del navegador y seleccionando la opción “Inspeccionar”. El desarrollador debe buscar el formulario correspondiente al caso prueba y debe dar clic en el botón “Seleccionar” de la extensión para que se dibuje un recuadro sobre el formulario, al dar clic en recuadro la extensión guardará todos los atributos de los inputs que encuentre

en el formulario y de esa forma se crea una copia del formulario.

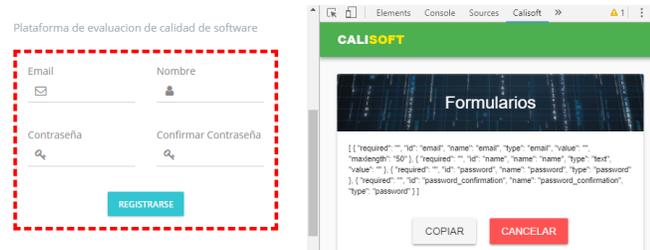


Fig. 4 Uso de la extensión de Calisoft para extraer los atributos de los inputs de un formulario destinado al registro de un usuario.

Cabe resaltar que la copia se almacena como un JSON.

Cuando el caso de prueba es actualizado con el formulario del desarrollador, el evaluador procede a realizar el testeo no heurístico.

Tipo	Nombre	Reglas	Input	Estado
email	email	email required max:50	<input type="text"/>	✘
text	name	required	<input type="text"/>	✘
password	password	required	<input type="password"/>	✘
password	password_confirmation	required	<input type="password"/>	✘

Fig. 5 Módulo de Testing. En este módulo se recrea la copia del formulario que un desarrollador sube a la plataforma, en el cual se procede a ejecutar las pruebas no heurísticas automatizadas para el proceso de evaluación de calidad.

El sistema realiza una interpretación de los inputs del formulario y únicamente del evaluador debe presionar un botón para que el sistema ingrese datos que obtiene desde la base de datos, esto se logra gracias a la herramienta proporcionada por el framework Laravel llamada; Laravel Faker, “Es un asistente que ingresa datos existentes en la base de datos que coinciden con un conjunto determinado de criterios”[9]. Cada sentencia que viene de la

base de datos tiene un atributo que se llama “valido”, de tipo booleano donde un 0 representa que la sentencia corresponde a un dato incoherente o malicioso, y un 1 representa que la sentencia está bien digitada para el input sobre el cual se va a probar y además no es una sentencia maliciosa para el sistema. Resalto que uso el término “Malicioso” para hacer referencia a una sentencia destinada a ataques de Sql o Xss que pueden averiar el funcionamiento del software si se le permite ingreso a la base de datos.

Por el lado del input que el sistema recrea, este viene acompañado de una serie de reglas, que son las mismas que el desarrollador cuenta en su aplicativo web.

Reglas	
email required max:50	<input type="checkbox"/>
required	<input type="checkbox"/>
required	<input type="checkbox"/>
required	<input type="checkbox"/>

+
CARGAR PRUEBA

Fig. 6 Reglas provenientes del formulario. Estas reglas se usan para corroborar la correcta validación que implementó el desarrollador en su formulario.

Estas reglas se combinan con un plugin de Vue.js llamado Vee-Validate, cuyo objetivo es validar campos de entrada y mostrar errores de una forma fácil y potente”[10]. Lo cual hace que recree esas mismas reglas sobre el input y que además si se dispara una de ellas, aparecerá un mensaje en de color rojo debajo el input.

Reglas	Input	Estado
email required max:50	<input --"="" ;="" drop="" table="" type="text" users;="" value="\"/> El campo email debe ser un correo electrónico válido.	✓
required	<input --"="" ;="" drop="" members;="" table="" type="text" value="\"/>	✗
required	<input type="text" value="....."/>	✗
required	<input type="text" value="....."/>	✗

Fig. 7 Plugin Vee-validate. El plugin se activa al momento en que permite el ingreso de información errónea o maliciosa.

El sistema cuenta con una programación booleana que al momento de ingresar un dato y si el sistema no muestra ningún error por medio del componente de javascript, almacenará un 0, en cambio, si el componente se dispara y muestra un error, el sistema almacenará un 1.

El método de calificación es el siguiente; se emplea una lógica simbólica denominada como “Disyunción exclusiva y operador OR exclusivo (XOR)”, se denota por el símbolo \oplus , donde un valor es verdadero cuando dos proposiciones q y p son contrarias y si q y p son dos proposiciones iguales, el valor obtenido es falso [11].

TABLA I
DISYUNCIÓN EXCLUSIVA Y OPERADOR OR EXCLUSIVO (XOR)

Sentencia traída de base de datos (p)	Error proveniente del componente de Javascript (q)	Calificación (p \oplus q)
0	1	1
1	0	1
0	0	0
1	1	0

Si la sentencia proveniente de la base de datos (p) es correcta, se denota por un 1, este valor

está almacenado en el campo “valido” que anteriormente se mencionó, si no se dispara ningún error al ingresar ese dato, el sistema registrará un 0 para ese error, que significa que no encontró un error en la sentencia para ese tipo de input, implementando XOR obtenemos como resultado un 1, que equivale a una calificación correcta. Esta calificación se puede apreciar en la columna “Estado” en la figura 7.

Por cada inserción de la base de datos sobre un input se aplica la lógica XOR para determinar si está bien o mal las validaciones que uso el desarrollador. Cada prueba tiene una calificación sobre el formulario, la calificación va ligada a la cantidad de inputs que el formulario posea, se toma un valor de 100% y se divide en la cantidad de inputs, eso le dará a cada input un porcentaje, por ejemplo, si un formulario tiene cuatro inputs, cada input tendrá un valor de 25%, si la calificación obtenida por la lógica de XOR es positiva, el valor se cuenta como un 25 entero, pero si es negativa, la calificación (nota) se toma como 0, al final se suman todos los resultados como se aprecia en la formula (1).

$$Nota = \sum_{i=0}^{CantidadInputs} \frac{100}{CantidadInputs} \times ResultadoXOR \quad (1)$$

Una vez obtenida la calificación parcial de cada prueba, se procede a realizar una sumatoria de cada calificación parcial y dividirla sobre el número total de calificaciones parciales que se hicieron un formulario en específico, de ese modo, se obtiene la calificación final para dicho formulario, este proceso corresponde a la formula (2).

$$NotaFinal = \frac{\sum CalificacionParcial}{CantidadDeCalificacionesParciales} \quad (2)$$

El valor obtenido es guardado en la tabla donde está creado el caso prueba del

correspondiente formulario en el apartado de “calificación”.

VII. CONCLUSIONES

- La automatización de las pruebas permite que haya un estándar en el proceso de evaluación, ya no es necesario la creatividad o experiencia del evaluador para probar múltiples opciones datos correctos e incorrectos sobre cualquier formulario.
- La calidad de evaluación aumentó, dado que se disminuyó tiempos de evaluación, tiempos de respuesta y se optimizaron procesos que anteriormente eran un poco largos y complejos.
- Para la elaboración de este módulo, se implementó dos plugin que son base en su funcionamiento, el plugin Vee.Validate de Vue.js, encargado de validar en tiempo real un input con base a las reglas de validación implementadas por el desarrollador, y el plugin Laravel Faker, encargado de ingresar información proveniente de la base de datos de manera automática, este proceso se realiza sobre los sobre los inputs obtenidos del formulario que ofrece el desarrollador.

VIII. REFERENCIAS

- [1] M. Bolaños-Pizarro and R. Vidal-Infer, A; Navarro-Molina, Carolina; Valderrama-Zuriañ, Juan Carlos; Aleixandre-Benavent, “Usabilidad: concepto y aplicaciones en las páginas web médicas,” *Papeles Médicos*, vol. Vol 16, Nr, no. 1, pp. 14–21, 2007.
- [2] M. E. López Inga and R. M. Guerrero Huaranga, *Modelo de inteligencia de negocios y analítica en la nube para pymes del sector retail en Perú*, vol. 12, no. 20. Universidad Cooperativa de Colombia, 2018.

- [3] LIONS J. L., “ARIANE 5 Failure - Full Report,” 1996. [Online]. Available: <http://sunnyday.mit.edu/accidents/Ariane5accidentreport.html>. [Accessed: 19-Apr-2018].
- [4] N. G. Leveson and C. S. Turner, “An Investigation of the Therac-25 Accidents,” *Computer (Long Beach Calif)*., vol. 26, no. 7, pp. 18–41, Jul. 1993.
- [5] Y. Fundación Universitaria Luis Amigó. Facultad de Ingenierías. and V. Vega, *PRUEBAS DE SEGURIDAD: ESTUDIO DE HERRAMIENTAS*, no. 17. Fundación Universitaria Luis Amigó, Facultad de Ingenierías, 2009.
- [6] M. L. Rojas-Montes, F. J. Pino-Correa, and J. M. Martínez, *Proceso de pruebas para pequeñas organizaciones desarrolladoras de software*, vol. 24, no. 39. Universidad Pedagógica y Tecnológica de Colombia, 2015.
- [7] G. Kaplan *et al.*, “Generación semi automática de casos de prueba a partir de escenarios,” May 2015.
- [8] M. A. Mascheroni, M. K. Cogliolo, and E. Irrazábal, “Automatización de pruebas de compatibilidad web en un entorno de desarrollo continuo de software,” Nov. 2016.
- [9] Laravel, “Database Testing - Laravel - The PHP Framework For Web Artisans,” 2016. [Online]. Available: <http://laravel.com/docs/master/databases>. [Accessed: 26-Apr-2018].
- [10] Vue.js, “Vee Validar.” [Online]. Available: <https://vee-validate.logaretm.com/>. [Accessed: 26-Apr-2018].
- [11] C. Marcial, R. Carmona, and R. T. Milenio, “Matemáticas I.”

3.1.4.CRONOGRAMA

Etapas proyecto	Estado	2017												2018		
		Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	Enero	Febrero	Marzo	Abril	Mayo	Junio			
Requerimientos	Terminado	Requerimiento														
Planeación	Sí	Planeación														
Elaboración	Sí	Elaboración														
Construcción de marco teórico	Terminado		Construcción													
investigación	Sí		investigación													
Elaboración	Sí		Elaboración													
Modelado	Terminado		Modelado													
Planeación	Sí		Planeación													
Investigación	Sí		Investigación													
Elaboración	Sí		Elaboración													
Construcción de diseño ingeniería	Terminado			Construcción de diseño ingeniería												
Elaboración	Sí			Elaboración												
Transferencia y difusión	Terminando						Transferencia									
Pruebas de Modulo	Terminado							Pruebas de								
Retroalimentación	Terminado							Retroalimentación								
Integración	Terminado								Integración							
Pruebas Generales	Terminado								Pruebas							
Construcción de capítulos	Terminado									Construcción						
Manuales	Terminado									Manuales						
Sustentación Final	Terminado													Sustentación		