



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN SOFTWARE

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN SOFTWARE**

**TEMA: DESARROLLO DE UNA APLICACIÓN WEB QUE MEJORE LA
GESTIÓN DE LOS PRODUCTOS EXPERIMENTALES EN INGENIERÍA DE
SOFTWARE APLICANDO EL PARADIGMA DE LÍNEA DE PRODUCTO
SOFTWARE EN EL GRUPO DE INVESTIGACIÓN GITBIO.**

AUTOR: QUISHPE GUANOTÁSIG, CÉSAR EDUARDO

DIRECTOR: PHD. ESPINOSA GALLARDO, EDISON GONZALO

LATACUNGA

2018

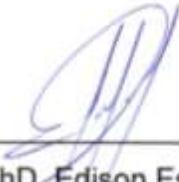


**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SOFTWARE**

CERTIFICADO DEL DIRECTOR

Certifico que el trabajo de titulación “DESARROLLO DE UNA APLICACIÓN WEB QUE MEJORE LA GESTIÓN DE LOS PRODUCTOS EXPERIMENTALES EN INGENIERÍA DE SOFTWARE APLICANDO EL PARADIGMA DE LÍNEA DE PRODUCTO SOFTWARE EN EL GRUPO DE INVESTIGACIÓN GITBIO” fue realizado por el señor **Quishpe Guanotásig, César Eduardo**, el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente

Latacunga, 31 de agosto del 2018



PhD. Edison Espinosa G.
C.C.: 0501577910
DIRECTOR DEL PROYECTO



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SOFTWARE

AUTORÍA DE RESPONSABILIDAD

Yo, **QUISHPE GUANOTÁSIG, CÉSAR EDUARDO**, con cédula de identidad N° 1311233710 declaro que el contenido, ideas y criterios del trabajo de titulación: **“DESARROLLO DE UNA APLICACIÓN WEB QUE MEJORE LA GESTIÓN DE LOS PRODUCTOS EXPERIMENTALES EN INGENIERÍA DE SOFTWARE APLICANDO EL PARADIGMA DE LÍNEA DE PRODUCTO SOFTWARE EN EL GRUPO DE INVESTIGACIÓN GITBIO”**, es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Latacunga, 31 de agosto del 2018

César Eduardo Quishpe Guanotásig

C.C.: 1311233710



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SOFTWARE**

AUTORIZACIÓN

Yo, **QUISHPE GUANOTÁSIG, CÉSAR EDUARDO**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“DESARROLLO DE UNA APLICACIÓN WEB QUE MEJORE LA GESTIÓN DE LOS PRODUCTOS EXPERIMENTALES EN INGENIERÍA DE SOFTWARE APLICANDO EL PARADIGMA DE LÍNEA DE PRODUCTO SOFTWARE EN EL GRUPO DE INVESTIGACIÓN GITBIO”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 31 de agosto del 2018

A handwritten signature in blue ink is positioned above a dashed horizontal line. The signature is cursive and appears to read 'César Eduardo Quishpe Guanotásig'.

César Eduardo Quishpe Guanotásig

C.C.: 1311233710

DEDICATORIA

Dedico este proyecto de investigación a mis queridos padres quienes, con su ejemplo de lucha, perseverancia y buenos valores, han sido mi guía y apoyo constante en cada transcurso de mi vida, sin ellos nada de esto hubiera sido posible. Ellos son mi fuente de motivación y pilares fundamentales.

A mis queridos abuelos Julio (+) y Josefina (+) quienes fueron mi ejemplo a seguir, siempre creyeron en mí y hoy me cuidan y me guían desde el cielo.

César Quishpe

AGRADECIMIENTO

Agradezco a Dios por brindarme paciencia, sabiduría y fortaleza para culminar con éxito mis metas propuestas.

A mis queridos padres César y Patricia, quienes me han apoyado incondicionalmente en todo momento, y me motivan a esforzarme por lograr las metas que me he propuesto.

A mi hermana Sandy, que con sus consejos me ha ayudado a afrontar los retos que se me han presentado en la vida.

Agradezco especialmente a mis tíos Pedro, Elizabeth, y Beatriz, quien con su ayuda comprensión y cariño han sido parte fundamental de mi vida.

A mi director de proyecto, Ing. Edison Espinosa PhD quien, con su experiencia, conocimiento y motivación me oriento en el desarrollo de la presente investigación.

César Quishpe

ÍNDICE DE CONTENIDOS

CARÁTULA

CERTIFICADO DEL DIRECTOR	ii
AUTORÍA DE RESPONSABILIDAD.....	iii
AUTORIZACIÓN	iv
DEDICATORIA.....	v
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDOS.....	vii
ÍNDICE DE TABLAS	x
ÍNDICE DE FIGURAS	x
RESUMEN.....	xii
ABSTRACT.....	xiii

CAPÍTULO I

INTRODUCCIÓN

1.1. Planteamiento del Problema.....	1
1.2. Antecedentes	2
1.3. Objetivo General	4
1.4. Objetivos Específicos.....	4
1.5. Justificación e Importancia.....	4
1.6. Hipótesis	5

CAPÍTULO II

MARCO TEÓRICO

2.1. Introducción	6
2.2. Experimentación en Ingeniería de Software	6
2.2.1. Ingeniería de Software Experimental	7
2.3. Proceso de Experimentación	8
2.3.1. Definición	8
2.3.2. Diseño.....	9

2.3.3. Ejecución	10
2.3.4. Análisis.....	10
2.3.5. Replicación de Experimentos.....	11
2.3.6. Problemática de Replicación en Ingeniería de Software.....	12
2.4. Línea de Producto Software.....	13
2.4.1. Características de la Línea de producto software	14
2.4.2. Beneficios de la LPS.....	16
2.4.3. Proceso para Desarrollar la LPS.....	18
2.4.4. Gestión de la configuración de la LPS	21
2.5. Técnicas de Modelado.....	22
2.5.1. Método FODA	22
2.5.2. Herramientas para modelar la Variabilidad de una LPS	24
2.6. Tecnología para el desarrollo de productos.....	28
2.6.1. Aplicación Web	28
2.6.2. HTML	28
2.6.3. Bootstrap.....	29
2.6.4. JavaScript	30
2.6.5. Firebase	30

CAPÍTULO III

DESARROLLO DE LA PROPUESTA

3.1. Diseño de una LPS	33
3.1.1. Análisis del Contexto.....	33
3.1.2. Registro Características Fase de Definición	33
3.1.3. Registro de Características de la Fase de Diseño	34
3.2. Modelado del Dominio	38
3.2.1. Modelo Genérico.....	38
3.2.2. Modelo Técnicas de Evaluación de Código	38
3.3. Implementación de LPS	40
3.3.1. Arquitectura.....	40
3.3.2. Configuración Firebase	40

3.3.3. Descripción de la interfaz.....	45
3.4. Limitaciones.....	64
3.5. Despliegue.....	65

CAPÍTULO IV

PRUEBAS DE LA APLICACIÓN

4.1. Introducción.....	68
4.2. Prueba de instanciación.....	68
4.2.1. Caso de Prueba HEIII31.....	69
4.2.2. Caso de Prueba HEIII312.....	71
4.2.3. Caso de Prueba HEIII313.....	73

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones.....	75
5.2. Recomendaciones.....	76

REFERENCIAS BIBLIOGRÁFICAS	77
---	----

ANEXOS	81
---------------------	----

ÍNDICE DE TABLAS

Tabla 1	Actividades del proceso de experimentación en IS	8
Tabla 2.	Características de LPS.....	14
Tabla 3	Beneficios de la LPS	16
Tabla 4.	Herramientas para modelar variabilidad.....	27
Tabla 5	Matriz Análisis de Contexto.....	34
Tabla 6.	Matriz Técnicas de Evaluación de Código	35
Tabla 7	Hipótesis Técnicas de Evaluación de código	68
Tabla 8	Prueba de Instanciación HEIII31	69
Tabla 9	Prueba de instanciación HEIII312	71
Tabla 10.	Prueba de instanciación HEIII313	73

ÍNDICE DE FIGURAS

Figura 1.	Actividades de la LPS.....	21
Figura 2.	Relaciones de FODA	24
Figura 3.	Modelo de Dominio para Experimentación	38
Figura 4.	Modelo de Dominio Técnicas de Evaluación de Código.....	39
Figura 5.	Reglas del Dominio.....	40
Figura 6.	Arquitectura de la Aplicación	40
Figura 7.	Pantalla Principal de Firebase	41
Figura 8.	Consola de Firebase.....	41
Figura 9.	Script Firebase.....	42
Figura 10.	Autenticación Firebase	42
Figura 11.	Almacenamiento Firebase	43
Figura 12.	CORS.json.....	44
Figura 13.	Configurar CORS.....	44
Figura 14.	Realtime Database Experimentación.....	45
Figura 15.	Pantalla de Ingreso.....	46
Figura 16.	Pantalla de registro de usuario	46
Figura 17.	Pantalla Principal	46
Figura 18.	Gestión de Experimentos	47
Figura 19.	Agregar Experimento.....	48
Figura 20.	Modificar Experimento	48
Figura 21.	Eliminar Experimento.....	49
Figura 22.	Gestionar Objetivos	49
Figura 23.	Agregar Objetivo.....	50
Figura 24.	Modificar Objetivo	51
Figura 25.	Gestionar Hipótesis	51

Figura 26. Agregar Hipótesis	52
Figura 27. Modificar Hipótesis	52
Figura 28. Gestionar Variable Respuesta	53
Figura 29. Agregar Variable Respuesta.....	53
Figura 30. Modificar Variable Respuesta	54
Figura 31. Gestionar Métrica	55
Figura 32. Agregar Métrica	55
Figura 33. Modificar Métrica	56
Figura 34. Gestionar Factores	57
Figura 35. Agregar Factor.....	57
Figura 36. Modificar Factores	58
Figura 37. Gestionar Instrumentos	58
Figura 38. Agregar Instrumentos.	59
Figura 39. Modificar Instrumento	59
Figura 40. Gestionar Materiales.....	60
Figura 41. Agregar Materiales	60
Figura 42. Modificar Material	61
Figura 43. Gestionar Objetos.....	61
Figura 44. Agregar Objetos.....	62
Figura 45. Modificar Objetos.....	62
Figura 46. Configurar Experimento.....	63
Figura 47. Productos Experimentales.....	64
Figura 48. Crear App Heroku.....	65
Figura 49. Panel de Administración Heroku.....	66
Figura 50. Iniciar sesión en Heroku	66
Figura 51. Iniciar Repositorio Git	66
Figura 52. Despliegue en Heroku	67
Figura 53. Instanciación HEIII31	70
Figura 54. Instanciación HEIII312.....	72
Figura 55. Instanciación HEIII313.....	74

RESUMEN

El presente trabajo de titulación propone el desarrollo de una aplicación web que permita la gestión de productos experimentales en Ingeniería de Software aplicando el paradigma Línea de Producto Software. Se denominan productos experimentales a los instrumentos, materiales y objetos que se utilizan en las fases de definición y diseño de un experimento, estos productos son esenciales para ejecutar una replicación y consolidar los resultados obtenidos del experimento, la experimentación en el área de Ingeniería de Software permite la generación de conocimiento validado científicamente sobre métodos, técnicas, herramientas entre otros que se aplica en el desarrollo de software, por tal motivo es importante poner a disposición de los investigadores interesados en replicar un determinado experimento, la información de los respectivos productos experimentales. Para aplicar la LPS en el proceso de experimentación se obtuvo la información de fuentes primarias, posteriormente se registró en la técnica de análisis de contexto. El diseño del dominio se realizó, aplicando el método FODA y el modelo se implementó con la herramienta FeatureIDE. La aplicación web se desarrolló utilizando el lenguaje Javascript y los servicios de Autenticación, Realtime Database y almacenamiento que provee la plataforma de Firebase. La propuesta se validó mediante pruebas de instanciación de los productos experimentales del caso práctico “Técnicas de evaluación de código”.

PALABRAS CLAVE:

- **APLICACIONES WEB**
- **INGENIERÍA DE SOFTWARE**
- **LÍNEA DE PRODUCTO SOFTWARE**

ABSTRACT

The present project proposes the development of a web application that allows the management of experimental products in Software Engineering that applies the Software Product Line (SPL) paradigm. Instruments, materials and objects are called experimental products that are used in the definition and design phases of an experiment, products and components for the implementation of a data analysis tool. allows the generation of scientifically validated knowledge about methods, techniques, tools among others, applied in software development, why it is important to put into practice those interested in replicating an analysis, the information of the names of experimental products. To apply the SPL in the experimentation process, information was obtained from primary sources, then registered in the context analysis technique. The design of the domain was carried out, applying FODA and the model is implemented with FeatureIDE tool. The web application was developed using Javascript and the services Authentication, Realtime Database and storage provided by the Firebase platform. The proposal was validated by instantiation tests of the experimental products in the case of study “Code evaluation techniques”.

KEY WORDS:

- **WEB APPLICATION**
- **SOFTWARE ENGINEERING**
- **SOFTWARE PRODUCT LINE**

CAPÍTULO I

INTRODUCCIÓN

1.1. Planteamiento del Problema

El conocimiento que se genera en ingeniería de software sobre métodos, técnicas, herramientas entre otros que se aplica en el desarrollo de software es empírico, debido a que éste es resultado de la experiencia y no de la aplicación del método científico (ingeniería del software experimental). Actualmente, la ISE aplica procesos de experimentación y replicación para evaluar productos intermedios y finales (ERS, diseños, código, pruebas, sistemas software entre otros) que se generan por la aplicación del proceso de desarrollo software.

La experimentación en ingeniería de software (IS) permite generar conocimiento científico que se utiliza para desarrollar software de calidad. Para validar este conocimiento se requiere de la replicación, que es repetir el experimento tantas veces como se pueda en busca de consolidar y validar los hallazgos encontrados y publicados sobre el experimento.

Desde décadas atrás en la IS se ejecutan algunos experimentos como los de revisión de código, ingeniería de requisitos, pruebas, usabilidad entre otros, donde se evalúan técnicas, métodos y herramientas que permitan desarrollar software.

Una de las carencias encontradas en los grupos de investigación en Ingeniería de Software es el soporte a las actividades de planificación y ejecución de experimentos o repeticiones. El análisis de la literatura sobre la temática relaciona a que podría ser

porque no se cuenta con una herramienta que automatice el proceso de gestión de productos experimentales.

Actualmente, la información que está disponible de los productos para los experimentadores es incompleta, dispersa y con un nivel de detalle bajo, debido al elevado número de versiones que existen de los productos experimentales que se encuentran almacenados en diferentes sitios, con distintos formatos y contenidos.

El no disponer de información eficiente, efectiva y en forma oportuna sobre el estado y la traza de los experimentos, replicaciones y materiales experimentales, son un limitante que imposibilita la toma de decisiones adecuadas a los experimentadores, lo que genera pérdida de tiempo, costos y recursos que se utilizan en los procesos de planteamiento, diseño y ejecución de experimentos y replicaciones.

Acorde a lo anteriormente indicado, se formula la interrogante: ¿Cómo mejorar la gestión de los productos experimentales en ingeniería de software aplicando el paradigma de línea de producto software en el grupo de investigación GITBIO?

1.2. Antecedentes

En Ingeniería de software la norma IEEE 12207 (IEEE, 1990) establece el proceso de gestión de configuración de software (GCS) para mitigar el caos provocado por las múltiples versiones de productos (ERS, código, bases de datos, interfaces entre otros) generados por la aplicación del proceso de desarrollo software. Para soportar el proceso de GCS la IEEE proporciona el estándar IEEE828 (ANSI/IEEE, 1983) donde se planifican y organizan las actividades para seleccionar, identificar y controlar el estado y traza de

los productos intermedios y finales. Estas actividades se realizan manualmente, por lo que para agilizar este proceso muchos investigadores desarrollaron proyectos software para automatizarlo y así permitir al equipo de desarrollo que sean conscientes del estado y traza de los productos software.

En Ingeniería de Software Experimental (ISE), existen grupos de investigación (GITBIO, GRISE y Modelos de producción de software) que tienen el problema de la gestión de las versiones de los productos experimentales (materiales, objetos, instrumentos entre otros) físicos y digitales. Las distintas versiones de los productos son el resultado de aplicación de experimentos y replicaciones para evaluar métodos, técnicas y herramientas que permitan desarrollar Software de calidad haciendo un uso eficiente de los recursos.

Actualmente, aplicaciones como Git, RationalClearCase, Subversion, Tortoise SVN, permiten realizar el control de versiones tanto de los productos software como de archivos digitales creados por distintas aplicaciones software (procesadores de texto, hojas de cálculo, bases de datos entre otras). Estos son almacenados en carpetas que están sujetas a control de cambios con alguna de las herramientas anteriormente indicadas. Esta gestión de cambio se realiza a partir de la fecha en que son incorporados los archivos a la GCS automatizada.

Los productos experimentales generalmente tienen nombres similares o diferentes, pero con contenido similar y han sido creados y/o modificados en diferentes fechas para diferentes propósitos. Las soluciones software de control de versiones antes indicadas no proveen de funcionalidades para el escenario antes descrito en ISE. Por lo que se

requiere implementar una herramienta que permita la gestión de los productos experimentales en Ingeniería de Software.

1.3. Objetivo General

Desarrollar una aplicación web que mejore la gestión de los productos experimentales en ingeniería de software aplicando el paradigma de línea de producto software en el grupo de investigación GITBIO.

1.4. Objetivos Específicos

- Elaborar el marco teórico que fundamente la gestión de productos experimentales y el proceso de línea de producto software.
- Diseñar e implementar la aplicación web de gestión de productos experimentales.
- Validar los resultados obtenidos en la aplicación web, con un caso de estudio práctico.

1.5. Justificación e Importancia

Actualmente la ISE utiliza el método científico para generar conocimiento sobre técnicas, métodos, herramientas entre otros que se utilizan para desarrollar software en base a la ejecución de experimentos y replicaciones. Éstos son realizados en varios grupos de investigación en diferentes universidades en el mundo.

Es importante técnicamente el desarrollo de una aplicación web para la gestión de los productos experimentales en ingeniería de software, porque facilitaría a los experimentadores interesados en replicar un experimento decidir que materiales deben

usar para un determinado contexto, lo cual mejoraría la fase de planificación de experimentos y replicaciones.

Económicamente el proyecto es trascendental porque permitirá reducir costos en el proceso de replicación puesto que, al automatizar la gestión de productos experimentales, se agiliza la configuración de nuevas replicaciones.

Socialmente es importante debido a que el proyecto incentivará a los miembros de los grupos de investigación (GITBIO, GRISE y Modelos de producción de software) a realizar experimentos y replicaciones en Ingeniería de Software, generando conocimiento científico que permita mejorar la calidad en el proceso de desarrollo de software.

1.6. Hipótesis

Si se desarrolla una aplicación web entonces se mejora la gestión de los productos experimentales en ingeniería de software aplicando el paradigma de línea de producto software en el grupo de investigación GITBIO.

CAPÍTULO II

MARCO TEÓRICO

2.1. Introducción

En el presente capítulo se detallan los fundamentos teóricos que ayudan a conceptualizar el proceso experimental en el área de Ingeniería en Software, se describe el paradigma LPS, sus beneficios y características, posteriormente se establece el proceso para desarrollar la LPS, y se especifican las herramientas para modelar la variabilidad de una LPS.

2.2. Experimentación en Ingeniería de Software

La Ingeniería en Software (IS) “es una disciplina joven cuyo objetivo principal es la aplicación de enfoques sistemáticos, disciplinados y cuantificables para diseñar, desarrollar, operar y mantener un producto software” (IEEE, 1990)

Generalmente, en cualquier ingeniería se recopila el conocimiento científico referente a un problema de dominio tecnológico de modo que este sea verdaderamente útil para el profesional, brindando así respuestas a preguntas que acontecen comúnmente en la práctica. (Shaw, 1990)

En IS no se tiene suficiente evidencia que soporte las prácticas utilizadas en el proceso de desarrollo de software. Debido a la falta de fundamentos conceptuales sólidos para la disciplina. (Wohlin, y otros, 1999)

Los resultados de la aplicación de una determinada tecnología para el desarrollo de software son actualmente impredecibles (Wohlin, y otros, 1999). Frecuentemente se

adoptan nuevas tecnologías sin la certeza de que estas serán realmente efectivas (Zelkowitz, Wallace, & Binkley, 2003).

En la actualidad aún se debe consolidar el conocimiento científico en IS de tal forma que el ingeniero de software pueda aplicarlos de forma efectiva. Una manera de alcanzar este objetivo es la aplicación del proceso experimental, también conocido como Ingeniería de Software Experimental. (Gómez, Ucán, & Gómez, 2013)

2.2.1. Ingeniería de Software Experimental

La Ingeniería de Software Experimental (ISE) integra el paradigma experimental con la Ingeniería en Software, este paradigma se ha aplicado de forma exitosa en disciplinas científicas clásicas como la medicina, la física, y la química, en la actualidad ha permitido progresar en disciplinas tradicionalmente no experimentales, tales como la economía o la psicología. (Fernández, 2012)

La ISE decreta los principios para ejecutar investigaciones experimentales en IS con el fin de obtener conclusiones científicamente válidas sobre las que basar el desarrollo de software. Este conocimiento debe permitir identificar las condiciones de aplicabilidad, las debilidades y las fortalezas de las distintas tecnologías de desarrollo de software. (Basili V. R., 1993)

El objetivo específico de la ISE es la generación de conocimiento de calidad científica, es decir, conocimiento empíricamente maduro (hechos) que contrarreste el conocimiento subjetivo (opiniones y suposiciones) manejado normalmente en el desarrollo de software. (Juristo & Moreno, 2001)

La unidad fundamental del paradigma experimental es el experimento. En un experimento se modelan las características esenciales de una realidad (desarrollo de software) bajo condiciones controladas, que permitan estudiar y comprender de manera óptima esa realidad. De tal manera, los experimentos son útiles para confirmar o refutar teorías, creencias (juicios convencionales), o explorar relaciones causales. (Gómez, Ucán, & Gómez, 2013)

2.3. Proceso de Experimentación

Varios investigadores determinaron los procesos para realizar experimentos en cuatro fases principales: definición, diseño o planificación, ejecución y análisis. (Gómez, Ucán, & Gómez, 2013)

En la Tabla 1 se muestran las actividades principales del proceso de experimentación desde la perspectiva de cada investigador.

Tabla 1
Actividades del proceso de experimentación en IS

Autor	Definición	Diseño	Ejecución	Análisis
Basili	Definición	Planificación	Operación	Interpretación
Wohlin	Definición	Planificación	Operación	Análisis e Interpretación
Juristo y Moreno	Objetivo	Diseño	Ejecución	Análisis

Fuente: (Gómez, Ucán, & Gómez, 2013)

2.3.1. Definición

Esta actividad consiste en especificar de forma general los aspectos fundamentales del experimento a realizar, definir las hipótesis de trabajo. En esta fase se puede utilizar

el método GQM (Goal-QuestionMetric) que permite delimitar: el objeto de estudio del experimento, el propósito, la perspectiva, el enfoque de calidad, y el contexto donde se realiza el experimento. (Gómez, Ucán, & Gómez, 2013)

El método GQM se utiliza para especificar el objetivo y las metas del experimento. (Wohlin, y otros, 1999). Por otra parte, se pueden definir las hipótesis a examinar a partir de la especificación del objetivo del experimento. (Juristo & Vega, 2009). Estos dos enfoques pueden complementarse para llevar a cabo esta actividad. (Gómez, Ucán, & Gómez, 2013)

2.3.2. Diseño

Al definir el experimento y sus hipótesis, la siguiente fase de este proceso consiste en diseñar cómo se llevará cabo el experimento. En esta actividad se especifica cómo se asignan los tratamientos a los sujetos, el tipo de sujetos a emplear, así como se preparan los instrumentos, materiales y objetos necesarios para ejecutar el experimento. Respecto a cómo asignar los tratamientos a los sujetos, en la literatura existen diferentes tipos de diseños experimentales que satisfacen propósitos particulares. (Gómez, Ucán, & Gómez, 2013)

2.3.2.1. Variables

Un experimento está compuesto por dos tipos de variables que son: factores y variables de respuesta. Los factores son aquellas variables que se pueden manipular o controlar en el experimento, mientras que las variables de respuesta son variables que

se analizan para observar el efecto que producen los cambios en los factores. (Gómez, Ucán, & Gómez, 2013)

2.3.2.2. Materiales experimentales

Los materiales experimentales son un conjunto común interrelacionado, reutilizable, almacenados en repositorios, que se utilizan para ejecutar futuros procesos de experimentación en el ciclo de investigación experimental. Estos materiales evolucionan en el tiempo debido a los cambios que se producen en los diferentes niveles del experimento, cambios como: del dominio del experimento, la temática del experimento, el problema, o de elementos concretos como modificación del diseño, debido al resultado de aplicar nuevas replicaciones para generar nuevo conocimiento validado (Gómez, Juristo, & Vegas, 2010).

2.3.3. Ejecución

Luego de definir el experimento y seleccionar el tipo de diseño, se procede a realizar el experimento. En esta fase se tienen listos los materiales a utilizar, los equipos, y el espacio físico donde realizará el experimento. Antes de comenzar la sesión del experimento, los sujetos que interviene en la ejecución reciben un conjunto de instrucciones, así como los materiales con los que trabajarán durante la sesión del experimento. (Gómez, Ucán, & Gómez, 2013)

2.3.4. Análisis

Esta actividad se realiza después de ejecutar el experimento. El investigador se encarga de recoger una serie de métricas generadas por los sujetos experimentales, estas métricas son analizadas y comparadas con las hipótesis preliminarmente definidas.

En el análisis se utilizan diferentes técnicas estadísticas que permitan el análisis e interpretación de las métricas obtenidas. Usualmente en el diseño de experimentos se emplea el análisis de la varianza (ANOVA). Esencialmente, ANOVA consiste en inspeccionar la variabilidad obtenida de las métricas en relación a los tratamientos aplicados, así como reconocer la variabilidad de otros tipos de variables. El ANOVA proporciona una prueba estadística sobre las medias o promedios de diversos grupos son o no equivalentes. (Gómez, Ucán, & Gómez, 2013)

2.3.5. Replicación de Experimentos

La realización de experimentos no constituye un fin en sí mismo. El experimento es parte de una cadena de preguntas y respuestas parciales sobre un fenómeno. Para poder profundizar en el conocimiento y obtener mayor confianza sobre los resultados, se deben replicar los experimentos (Solari Buela, 2011)

La replicación es una noción esencial en el proceso experimental. No es posible imaginar una ciencia experimental donde no se repliquen los resultados. El método para la evolución de la evidencia y aclaración de las relaciones causales de un fenómeno es la replicación. No obstante, las distintas definiciones existentes en la replicación difieren en el alcance del concepto. Hay varias formas de representar las replications. Esto sucede también en disciplinas científicas más maduras que la ingeniería de software. (Gómez, Juristo, & Vegas, 2010).

Actualmente hay varias formas de replicar un experimento. Una forma de replicación consiste en que el propio investigador repita el experimento para afirmar que el resultado obtenido no fue producto de la casualidad. Posteriormente, se puede necesitar que otros

investigadores participen, para asegurarse que las observaciones realizadas no dependen de la persona que las realizó. (Brooks, Daly, Miller, Roper, & Wood, 1996).

2.3.6. Problemática de Replicación en Ingeniería de Software

Es complicado realizar replications en IS, el motivo principal es la inmadurez del paradigma experimental en el proceso de desarrollo de software. Actualmente, los resultados que se obtienen de aplicar el proceso de replicación originan resultados experimentales distintos y difíciles de conciliar con el experimento original, debido al desconocimiento de la tipología de sujetos y tareas, características de los problemas (Juristo & Moreno, 2001). Estos factores pueden ocasionar un cambio que afecta, a propósito, o inadvertidamente, a la replicación, causando que los resultados del experimento original sean imposibles de replicar (Fonseca, 2012).

La realización de una replicación ejecutada por un grupo distinto de investigadores al que efectuó el experimento conlleva una serie de desafíos. Primero, se debe tener en cuenta la necesidad de transferencia del conocimiento experimental. (Shull, y otros, 2004).

Para realizar la replicación de un experimento, la comunicación entre los miembros del grupo de investigadores es importante. Existen diversas formas de comunicación, como interacción directa entre los investigadores y la documentación del experimento. Los materiales experimentales pertenecen al último grupo. El material experimental ideal es aquel que contuviera todo el conocimiento que sea necesario para replicar un experimento (Solari Buela, 2011). Sin embargo, actualmente se carece de información eficiente, efectiva y en forma oportuna sobre el estado de los experimentos, y los

materiales experimentales, generando, pérdida de tiempo, costos y recursos que se utilizan en los procesos de planteamiento, diseño y ejecución de experimentos y replicaciones, imposibilitando la toma de decisiones adecuadas a los experimentadores. (Espinosa, 2014)

2.4. Línea de Producto Software

Clements y Northrop definen a una Línea de Producto Software (LPS), como: “un conjunto de sistemas software, que comparten un conjunto común de características (features), las cuales satisfacen las necesidades específicas de un dominio o segmento particular de mercado, y que se desarrollan a partir de un sistema común de activos base (core assets) de una manera preestablecida”. (Clements & Northrop, 2001)

Vinga-Martins manifiesta que: “Una línea de productos software es un conjunto de soportes lógicos que poseen unas características comunes (features) y que están orientados a las necesidades específicas de una determinada área de negocio”. (Vinga-Martins, 1999).

Según Charles Krueger: “LPS se refieren a técnicas de ingeniería para crear un portafolio de sistemas de software similares, a partir de un conjunto compartido de activos de software, usando un medio común de producción”. (Krueger, 2006)

Gomaa conceptualiza de una forma simple y concreta a la LPS: “consiste de una familia de sistemas de software que tienen una funcionalidad común y alguna funcionalidad variable”. (Gomaa, 2004)

En base a las definiciones anteriores se puede apreciar que existen dos nociones fundamentales para la aplicación de LPS:

- **Feature:** es una característica conceptual de un sistema y se utiliza para describir o distinguir un producto dentro de la línea, pueden relacionarse con características visibles de los sistemas, otras con la estructura de los sistemas, e incluso podrían describir requisitos no funcionales.
- **Core asset:** es un artefacto de software que se utiliza en el proceso de desarrollo de productos de la LPS, pueden ser: componentes software, modelos de procesos, arquitectura, documentos, etc. Se le considera como el núcleo principal de todos los productos que están dentro de la línea de producto. (Bastarrica, 2002)

2.4.1. Características de la Línea de producto software

Según Bastarrica (Bastarrica, 2002), LPS tiene características únicas las cuales se describen a continuación:

Tabla 2.
Características de LPS

Item	Características
Creación de core assets	Se crea un conjunto de elementos comunes (core assets) con el fin de ser reutilizados en la creación de un conjunto de productos que forman parte de una línea de producto. Los core assets se reutilizan de una manera planificada en LPS. La planificación indica la forma en que se deben utilizar los core assets para crear los distintos productos de la LPS.



Utilización de core assets	No necesariamente todos los core assets son utilizados para la creación de los productos de la línea. No obstante, cada core asset se utiliza para la creación de la mayor cantidad de productos posibles de la línea de producto.
Creación de Productos	Todos los productos de la línea se crean desde la misma base de core assets. Para ellos se tiene conocimiento de cuáles son los elementos comunes y variables dentro de la línea.
Trato de la LPS	La LPS se trata como un todo y no como un conjunto de sistemas que se mantienen por separado.
Arquitectura	La LPS dispone de una arquitectura re-configurable que se denomina arquitectura de referencia. La arquitectura de referencia es sólo una parte (quizá la más importante) de los principales core assets. Todos los productos de la línea van a compartir la arquitectura y por esta razón una de las actividades importantes en LPS es obtener la arquitectura de referencia la cual permite modelar las variaciones de los productos de la LPS.
Versiones	En LPS se dispone de distintas versiones liberadas de productos individuales para su uso. El proceso de LPS permite crear muchos productos simultáneamente en base a la utilización de los core assets.

CONTINÚA 

Actividades en LPS	Las actividades que se realizan en LPS requieren de la aplicación de estándares técnicos que permitan asegurar la calidad de los productos de la línea.
Fase Final	No tiene una fase final ya que extiende su vida útil en el tiempo en base a los productos a construir.
Actividades en los proyectos	Son interactivas y demanda mucha coordinación por parte de los administradores y el equipo de desarrollo de software.

2.4.2. Beneficios de la LPS

Una LPS puede generar diversos beneficios en lo organizacional, comercial y en ingeniería de software como se muestra en la Tabla 3

Tabla 3

Beneficios de la LPS

BENEFICIOS DE UNA LPS	
Beneficios organizacionales	<ul style="list-style-type: none"> • Provee ventajas grupales como una mejor comprensión de dominio • Facilidad de formar personas • Producto de alta calidad • Confianza del cliente.

CONTINÚA 

Beneficios de ingeniería de software

- Reutilización de requisitos y sus componentes,
- Mejor análisis de los requisitos
- Otra visión sobre los requisitos para el cliente.
- Control de la calidad del software
- Establecimiento de estándares de programación
- Una forma de encontrar y borrar implementaciones redundantes y completar documentaciones reutilizables.

Beneficios comerciales

- Reducción del mantenimiento y los costos de producción (gracias a la reutilización de puntos en común entre varios productos).
- Generan una mayor eficiencia en el proceso de planificación y la posibilidad para mejorar el presupuesto y la planificación del tiempo.

2.4.3. Proceso para Desarrollar la LPS

El proceso incluye tres actividades que están en constante ejecución y retroalimentación, y que no tienen un orden establecido para su ejecución. Según Clements (Clements, Jones, Northrop, & McGregor, 2005), las actividades del paradigma LPS son: ingeniería del dominio (desarrollo de core assets), ingeniería de la aplicación (desarrollo de productos) y gestión del proceso de la LPS. (Espinosa, 2014)

2.4.3.1. Desarrollo de core assets

Se conoce como Ingeniería de dominio, esta actividad permite la identificación, recolección, organización y representación de la información esencial para el dominio de la LPS. En el análisis de dominio se describen las siguientes fases: análisis del contexto, modelado del dominio y modelado de la arquitectura de la LPS. (Kang, Cohen, Hess, Novak, & Peterson, 1990).

El análisis de contexto permite definir el ámbito del dominio de la LPS. Este se representa basado en la participación de los usuarios expertos en el contexto o dominio del sistema y el equipo de desarrolladores. Concretamente en esta fase, los desarrolladores se encargan de recoger la información de varias fuentes primarias como: artículos, opiniones de expertos, estándares, libros, etc. Una vez ejecutada esta fase se obtiene como salida un conjunto de requisitos expresados en forma de características (features). Estas características permiten diferenciar a un producto dentro de la LPS, y se emplean para desarrollar los core assets que representan las características comunes de los productos. En esta fase se emplean técnicas propias del desarrollo software como:

diagramas de contexto, diagramas de estructuras, diagramas de casos de uso, etc. (Salguero Espinosa & Salguero Espinosa, 2015)

La descripción de los problemas se realiza en base a las features dentro del dominio y son direccionadas para modelar la LPS, y permiten el diseño de algunos artefactos del dominio. En esta fase el equipo de desarrolladores modela el diseño genérico de la LPS, que se implementa basado en los requisitos del dominio generados en la fase de análisis del dominio. Además, el diseño genérico requiere que se realicen varias iteraciones entre usuarios y desarrolladores. Para modelar el dominio de la LPS se utiliza el modelo de características. (Salguero Espinosa & Salguero Espinosa, 2015)

La fase de modelado de la arquitectura permite implementar los componentes de código para brindar soporte a la arquitectura del dominio y posteriormente se reutilizarán durante la fase de desarrollo de productos que constituyen parte de la LPS. En el modelado de la arquitectura se utilizan técnicas como: modelo de procesos de iteración, modelo gráfico de estructura, modelo de componentes, etc.

(Kang, Cohen, Hess, Novak, & Peterson, 1990)

2.4.3.2. Desarrollo de productos

Es conocida como Ingeniería de la aplicación y permite el desarrollo de cada uno de los productos de la línea de producto (Bastarrica, 2002). Los productos que integran la LPS se implementan en esta actividad. Estos productos se construyen utilizando los requisitos específicos de cada uno de los productos y los productos resultantes de la ejecución de la ingeniería del dominio (Clements, Jones, Northrop, & McGregor, 2005).

LPS es un paradigma interactivo, en caso de no poder implementar los productos de la LPS partiendo de los core assets creados, permite regresar a la actividad de desarrollo de core assets.

Las fases de la ingeniería de la aplicación o desarrollo de productos son:

- a) Fase de configuración: esta fase se configuran los diferentes productos de la LPS. Esta configuración se ejecuta en base a la instanciación de los core assets y la selección de las features en los diferentes puntos de variación del producto. Se describen los puntos de variación en forma de requisitos específicos del producto de la LPS. Estos requisitos a construir están relacionados de alguna manera con los requisitos de la LPS a implementar.
- b) Fase de composición: en esta fase el equipo de desarrolladores implementan el código de los features. Las features facilitan distinguir a un producto de otro dentro de la LPS. Se obtiene como salida un producto configurado y desarrollado con características comunes y específicas de la LPS. (Espinosa, 2014)

2.4.3.3. Gestión del Proceso de la LPS

Esta actividad permite controlar y coordinar las distintas tareas o fases que se realizan en el transcurso del desarrollo de los core assets y de los productos de la LPS. Concretamente, admite la gestión de la variabilidad de los productos desarrollados en base a los core assets y los features (Bendix, Gradén.J, & Stahl, 2009). Adicionalmente,

se encarga de la coordinación del correcto funcionamiento de los equipos de trabajo de la LPS.

En la Figura 1 se puede apreciar tres círculos girando. Cada uno de los círculos representa una de las actividades del paradigma LPS. Las actividades son interactivas, se encuentran fuertemente relacionadas y están en constante desarrollo.



Figura 1. Actividades de la LPS.

Fuente: (Clements & Northrop, 2003)

2.4.4. Gestión de la configuración de la LPS

Debido a que el proceso para desarrollar la LPS es interactivo, se realiza una retroalimentación basada en el desarrollo de los core assets, las características específicas del producto y los productos de la línea. Cada una de estas actividades de la LPS posee un conjunto determinado de fases. Las fases en LPS permiten a los desarrolladores conocer las tareas a realizar y los productos resultantes de la aplicación de las mismas. (Espinosa, 2014)

En esta actividad se asignan, coordinan y supervisan recursos, tanto a nivel técnico (gestión de proyectos concretos) como organizativo (estructura organizativa adecuada a los objetivos propuestos) (Clements & Northrop, 2001). Además, esta actividad permite el control del estado y la traza de los core assets, features y productos de la LPS, esto permite que estén disponibles en forma fiable y oportuna para la creación de productos de la LPS. La actividad de gestión de la LPS al igual que las dos actividades descritas anteriormente exige mucha interacción y esfuerzo por parte de los usuarios, desarrolladores y especialmente a los administradores de la LPS.

2.5. Técnicas de Modelado

En la actualidad hay diversas técnicas para modelar características comunes y variables de una LPS. Uno de los primeros métodos propuestos fue FODA (Análisis de Dominio Orientado a características), a partir del cual se han derivado técnicas como FORM, y FeatureRSB. (Kang, Cohen, Hess, Novak, & Peterson, 1990)

2.5.1. Método FODA

Este método está formado por dos elementos: las características y sus relaciones. Las características se organizan en una estructura de árbol jerárquico. Una característica importante del árbol es la raíz, esta representa el sistema como un todo. Las relaciones entre características pueden ser de dos tipos: jerárquicas y no jerárquicas. (Salguero Espinosa & Salguero Espinosa, 2015)

2.5.1.1. Relación jerárquica

Se define entre una característica padre y sus características hijas. Una característica hija puede formar parte únicamente de los productos en los que la característica padre aparece. (Espinosa, 2014)

En FODA existen cuatro tipos de relaciones jerárquicas y se describen a continuación:

- a) Obligatoria (mandatory): indica que cuando la característica padre forma parte de un producto particular la característica hija también debe formar parte del producto.
- b) Opcional (optional): muestra que cuando la característica padre es parte de un producto particular, la característica hija, puede o no, formar parte del producto
- c) Alternativa (alternative): es la relación entre una característica padre y un conjunto de características hijas que indica que cuando la característica padre es parte de un producto particular, sólo una de las características del grupo de hijas debe ser parte del producto
- d) Or: indica que cuando la característica padre forma parte de un producto particular, una o más de sus características hijas debe ser parte del producto.

2.5.1.2. No jerárquica

Sirve para identificar las relaciones de composición de características y se establecen las reglas para desarrollar los productos de la LPS. (Espinosa, 2014)

- a) Excluye (excludes): Una característica X excluye Y significa que la característica X es incluida en el producto, la característica Y no debe ser incluida y viceversa.

b) Requiere (requires): Una característica X requiere Y significa que, si la característica X es incluida en el producto, la característica Y debe ser incluida y no viceversa.

2.5.1.3. Simbología

La simbología del método FODA permite representar un modelo jerárquico estructurado en niveles mediante las relaciones jerárquicas y no jerárquicas. En la figura 2 se aprecia la simbología propuesta por FODA para representar las relaciones utilizadas para modelar el dominio de la LPS. (Espinosa, 2014)

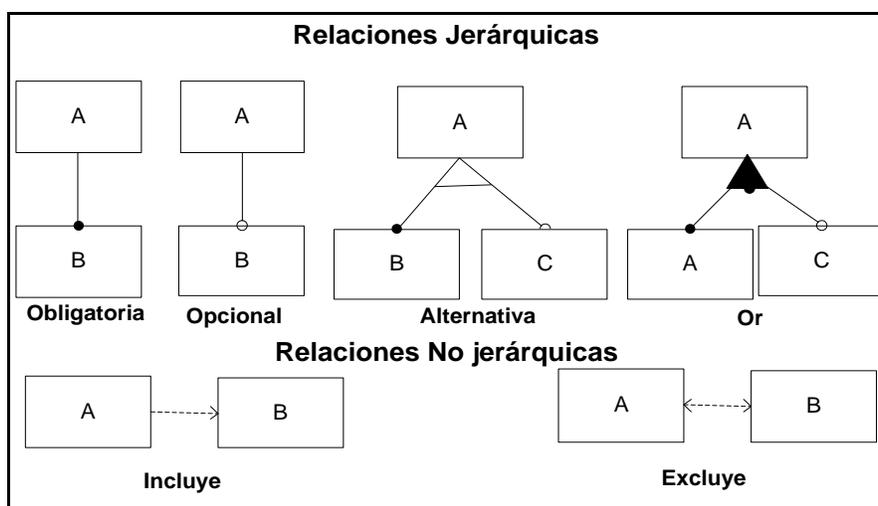


Figura 2. Relaciones de FODA

Fuente: (Kang, Cohen, Hess, Novak, & Peterson, 1990)

2.5.2. Herramientas para modelar la Variabilidad de una LPS

Actualmente se dispone de una variedad de herramientas para modelar la variabilidad de una LPS, las cuales permiten implementar modelos de características, configuraciones a través de una interfaz gráfica, y validar los modelos.

2.5.2.1. FeatureIDE

Es un framework de código abierto basado en Eclipse IDE que soporta el ciclo de vida de una Línea de Producto; análisis y modelado de dominio a través de un editor gráfico y de texto para modelar las características y restricciones en árboles jerárquicos (FODA), adicionalmente cubre el diseño, implementación y mantenimiento en el desarrollo de Software Orientado a características (FOSD). (Kästner, y otros, 2009)

Permite la integración con FeatureHouse, Feature C++, y AHEAD, como herramientas de composición y compiladores entre otros. Es un editor gráfico y textual que admite configurar y crear restricciones avanzadas. Utiliza como resolutor SAT. Se puede citar como desventajas que no brinda la función de características clonables, ni permite referencias entre características, y no dispone de atributos. (Salazar, 2009)

2.5.2.2. S.P.L.O.T

Es una aplicación web Opensource desarrollada por la Universidad de Waterloo de Canadá, está orientada a la utilización por parte de académicos y practicantes del paradigma LPS, entregando una herramienta poderosa que permite analizar y configurar modelos de decisión basados en características de los productos y modelándolas a través de cláusulas lógicas. (Urroz Urzúa, 2012)

Esta herramienta contiene implementados eficientes sistemas de razonamiento y heurísticas de resolución de problemas SAT, lo que permite contar configuraciones válidas y calcular el grado de variabilidad de las características de los productos. (Urroz Urzúa, 2012)

Actualmente S.P.L.O.T. muestra la profundidad de los árboles de características, cantidad de características, la cantidad de configuraciones permitidas y el grado de variabilidad del modelo. Con respecto a la configuración del producto, el sistema cuenta con la funcionalidad de generar una configuración de un producto de manera interactiva a través de la toma de decisiones por parte de los usuarios y la autoconfiguración que genera el sistema propagando las reglas definidas. (Urroz Urzúa, 2012)

No obstante, Splot no permite guardar localmente el proyecto creado de una línea de producto, no ofrece a los desarrolladores funcionalidades que permitan identificar similitudes y diferencias entre productos de una determinada línea. (Mendonça, Branco, & Cowan, 2009)

2.5.2.3. Pure Variants

Pure::Variants fue desarrollado por pure-systems como complemento de Eclipse, y es una herramienta de tipo propietaria. Brinda un conjunto integrado de herramientas que cubran cada una de las fases del desarrollo de la LPS, está disponible en la plataforma web de pure-systems, el acceso es limitado, se puede solicitar una licencia de prueba para la versión de evaluación, o a su vez se puede utilizar la versión Community con límite de modelamiento de hasta 100 características. (PURE::SYSTEMS, 2013)

Pure::Variants es un framework que permite la integración con varias herramientas y tipos de datos como: gestión de requisitos, análisis y diseño orientado a objetos, gestión de configuración, generadores de código, compiladores, código fuente, etc.

Pure::Variants proporciona las funcionalidades para soportar las actividades de ingeniería de dominio y aplicación de la LPS. (PURE::SYSTEMS, 2013)

Comparación de Herramientas de modelado

El análisis de herramienta para modelar la variabilidad de una LPS se realiza mediante una comparación entre FeatureIDE, S.P.L.O.T y Pure::Variants basado en los parámetros del artículo: "Análisis de desarrollo de software orientado a Feature - Línea De Producto de Software para Aplicaciones de TVDI". (Miranda, Casas, & Marcos, 2015)

Tabla 4.
Herramientas para modelar variabilidad

Descripción	S.P.L.O.T	FeatureIDE	Pure::Variants
Licencia	OpenSource	OpenSource	Comercial
Desarrollador	University of Waterloo Canadá	University of Magdeburg Alemania	Pure::systems
Plataforma	Web	Plugin Eclipse IDE	Plugin Eclipse IDE
Usabilidad	Alta	Media	Media
Editor de Modelo de Features	Si	Si	Si
Análisis del Modelo	Si	Si	Si
Configuración de Productos	Si	Si	Si
Notación del modelo	Árbol	Árbol	Árbol
Código del Modelo	XML	SXML	-

2.6. Tecnología para el desarrollo de productos

El producto a desarrollar es una aplicación web, actualmente existen varias tecnologías de desarrollo web, a continuación, se describen las nociones básicas de las tecnologías que se utilizarán para el desarrollo de este proyecto:

2.6.1. Aplicación Web

Una aplicación web es un conjunto de las herramientas que los usuarios pueden utilizar a través de internet o de una intranet, mediante un navegador web. En otros términos, es una aplicación software que se programa en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. La popularidad de estas aplicaciones surgió debido a lo práctico de un navegador web como cliente ligero, este es independiente del sistema operativo, permite realizar actualizaciones de manera fácil y el mantenimiento se realiza sin distribuir e instalar software a miles de usuarios potenciales. (Molina Caballero & Baena Espejo, 2007)

2.6.2. HTML

HTML ("HyperText Markup Language") es un lenguaje de marcado de hipertexto que permite estructurar el contenido de una página web: texto, imágenes, videos, entre otros mediante etiquetas o tags.

HTML es un lenguaje que interpreta el navegador web y permite al usuario visualizar los sitios o aplicaciones web.

Ventajas:

- Sencillo que permite describir hipertexto.

- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web o WYSIWYG.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.

Desventajas:

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- El diseño es más lento.
- Las etiquetas son muy limitadas.

2.6.3. Bootstrap

Es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales. Bootstrap fue desarrollado por Mark Otto y Jacob Thornton de Twitter, como un marco de trabajo (framework) para fomentar la consistencia entre las herramientas internas. Antes de Bootstrap, se usaban varias librerías para el desarrollo de interfaces de usuario, las cuales llevaban a inconsistencias y a una gran carga de trabajo en su mantenimiento. (Bootstrap, 2017)

2.6.4. JavaScript

Es un lenguaje de programación interpretado, definido como orientado a objetos, se basa en prototipos, imperativo, débilmente tipado y dinámico. Principalmente se usa en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. (Eslava Muñoz, 2012)

Se diseñó con una sintaxis similar al C, sin embargo, adopta nombres y convenciones del lenguaje de programación Java. Aunque Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes. (Eslava Muñoz, 2012)

Todos los navegadores actuales interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM). (Eslava Muñoz, 2012)

2.6.5. Firebase

Firebase es una plataforma de Google, que permite desarrollar aplicaciones móviles y web sin la necesidad de escribir código del lado del servidor, dispone de un conjunto de servicios para el desarrollo de como: autenticación, almacenamiento y base de datos, cabe recalcar que cuenta con más servicios como notificaciones, y analytics. (Firebase, 2016)

Para la presente investigación se utilizará los primeros 3 servicios y se describen a continuación:

2.6.5.1. Realtime Database

Firebase Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado.

Cuando compilas apps multiplataforma con los SDK de iOS, Android y JavaScript, todos los clientes comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes. (Firebase, 2016)

Firebase Realtime Database permite compilar aplicaciones ricas y colaborativas, ya que permite el acceso seguro a la base de datos directamente desde el código del cliente. Los datos persisten de forma local. Además, incluso cuando no hay conexión, se siguen activando los eventos en tiempo real, lo que proporciona una experiencia adaptable al usuario final. Cuando el dispositivo vuelve a conectarse, Realtime Database sincroniza los cambios de los datos locales con las actualizaciones remotas que ocurrieron mientras el cliente estuvo sin conexión, lo que combina los conflictos de forma automática. (Firebase, 2016)

Realtime Database es una base de datos NoSQL y, como tal, tiene diferentes optimizaciones y funcionalidades en comparación con una base de datos relacional. La API de Realtime Database está diseñada para permitir solo operaciones que se puedan ejecutar rápidamente. Eso permite crear una excelente experiencia de tiempo real que puede servir a millones de usuarios sin afectar la capacidad de respuesta. (firebase, 2016)

2.6.5.2. Autenticación

Este servicio permite a los usuarios autenticarse a través de proveedores de acceso social, tales como: Google, Facebook, Twitter, GitHub. Además, incluye un sistema de gestión de usuarios mediante el cual los desarrolladores pueden habilitar la autenticación

de usuario con el correo electrónico y la contraseña de inicio de sesión almacenado con Firebase. (Firebase, 2016)

2.6.5.3. Cloud Storage

Es un servicio de almacenamiento de objetos potente, simple y rentable construido para la escala de Google. Los SDK de Firebase para Cloud Storage agregan la seguridad de Google a las operaciones de carga y descarga de archivos para las aplicaciones de Firebase, sin importar la calidad de la red. Se puede utilizar el SDK para almacenar imágenes, audio, video, documentos y otros tipos de contenido generado por el usuario. En el servidor, se puede utilizar Google Cloud Storage para acceder a los mismos archivos. (Firebase, 2016)

Cloud Storage almacena los archivos en un depósito de Google Cloud Storage y los hace accesibles a través de Firebase y Google Cloud. Esto permite tener la flexibilidad para subir y descargar archivos de clientes móviles a través de los SDK de Firebase y realizar procesamiento en el servidor, como el filtrado de imágenes o la transcodificación de video mediante Google Cloud Platform. (Firebase, 2016)

Cloud Storage se escala automáticamente, lo que significa que no es necesario migrar a ningún otro proveedor. (Firebase, 2016)

CAPÍTULO III

DESARROLLO DE LA PROPUESTA

3.1. Diseño de una LPS

3.1.1. Análisis del Contexto

El análisis del Contexto de la Gestión de Materiales experimentales se fundamenta en la información presentada en la tesis “Gestión de configuración y línea de productos para mejorar el proceso experimental en ingeniería del software”. (Espinosa, 2014).

3.1.2. Registro Características Fase de Definición

- El planeamiento relaciona las características: objetivo, hipótesis, variable respuesta, métrica, factores.
- El objetivo puede ser de tipo general y específico, se debe registrar el objetivo general y además al menos un objetivo específico del experimento.
- La hipótesis. Se debe registrar al menos una hipótesis nula y una alternativa del experimento original.
- Se debe registrar al menos una variable respuesta del experimento original.
- Se debe registrar al menos una métrica que se relaciones con la variable respuesta del experimento.
- Se debe registrar al menos un factor del experimento.

3.1.3. Registro de Características de la Fase de Diseño

La fase de diseño asocia las características: instrumentos, materiales experimentales, y objetos.

- Se debe registrar la información de los instrumentos experimentales que se utilizan la ejecución del experimento.
- Se debe registrar la información de los materiales experimentales que utilizan en la ejecución del experimento original.
- Se debe registrar la información de los objetos experimentales que se utilizan en la ejecución del experimento original (Espinosa, 2014).

Tabla 5

Matriz Análisis de Contexto

Línea de Producto para Experimentación		
Fase de Definición		
	Característica	Valor Característica
Todo experimento debe tener al menos un objetivo.	Objetivo	Objetivo 1 Objetivo n
Todo experimento debe tener al menos una hipótesis nula y una alternativa.	Hipótesis	Hipótesis 1 Hipótesis 2 Hipótesis n
Todo Experimento debe tener al menos una variable respuesta	Variable Respuesta	Variable Respuesta 1 Variable Respuesta 2 Variable Respuesta n
Toda variable respuesta debe tener al menos una métrica.	Métrica	Métrica 1 Métrica 2 Métrica n
Todo experimento debe tener al menos un factor	Factores	Factor 1 Factor 2 Factor n

CONTINÚA 

Fase de diseño		
Todo experimento debe disponer de todos los instrumentos experimentales.	Instrumentos	Instrumento 1 Instrumento 2 Instrumento n
Todo experimento debe disponer de todo el material experimental.	Materiales	Material 1 Material 2 Material n
Todo experimento debe disponer de todos los objetos experimentales.	Objetos	Objeto 1 Objeto 2 Objeto n

La tesis doctoral: “Gestión de configuración y línea de productos para mejorar el proceso experimental en ingeniería del software”, sirvió como fuente primaria para obtener información de los materiales que se utilizaron para ejecutar el experimento “Técnicas de Evaluación de Código “.

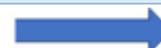
En base a la Tabla 5 se realizó una adaptación para el experimento Técnicas de Evaluación de Código y se describe los elementos que intervienen en las fases de definición y diseño experimental.

Tabla 6.

Matriz Técnicas de Evaluación de Código

Técnicas de Evaluación de código		
Fase de Definición		
Característica	Código	Descripción
Objetivo General	O1	Estudiar la efectividad y eficiencia de distintas técnicas de evaluación de código de código

CONTINÚA



Objetivo Específico	O2	Estudiar la efectividad y de clase de equivalencia, cobertura de sentencias y lectura de código
Hipótesis 1	HEIII31	Hay diferencia en la efectividad entre las técnicas de partición de clases de equivalencia y cobertura de sentencias
Hipótesis 2	HEIII312	Hay diferencia en la efectividad entre las técnicas de partición de clases de equivalencia y lectura de código
Hipótesis 3	HEIII313	Hay diferencia en la efectividad entre las técnicas de cobertura de decisión y lectura de código
Variable Respuesta	-	Efectividad
Métrica 1	MEIII1	Porcentaje de sujetos que generaron casos de prueba que detectaron fallas mediante técnica de evaluación dinámicas
Métrica 2	MEIII2	Porcentaje de sujetos que detectaron falla mediante técnicas estáticas
Métrica 3	MEIII3	Tiempo de generación de los casos de prueba
Métrica 4	MEIII4	Tiempo de ejecución de los casos de prueba
Métrica 5	MEIII5	Tiempo para detectar los fallos
Factor 1	Lectura de código	Step_Wise Abstraction
Factor 2	Caja Blanca	Cobertura de Decisiones
Factor 3	Caja Negra	Particiones de equivalencia
Fase de Planteamiento		
Instrumento 1	E12	Formulario de abstracciones para la técnica Revisión de Código

CONTINÚA



Instrumento 2	E13	Formulario de inconsistencias para la Revisión de Código
Instrumento 3	E22	Formulario de Datos de Prueba para las Pruebas Estructurales
Instrumento 4	E32	Formulario de Clases de Equivalencia para las Pruebas Funcionales
Instrumento 5	E33	Formulario de Datos de Prueba para las Pruebas Funcionales
Material 1	E10	Instrucciones para aplicar la técnica Revisión de Código
Material 2	E20	Instrucciones para aplicar la técnica Pruebas Estructurales
Material 3	E30	Instrucciones para aplicar la técnica de Pruebas Funcionales
Objeto 1	E01	Especificación del programa
Objeto 2	E02	Descripción del programa para la aplicación de la técnica
Objeto 3	E03	Descripción del programa para la aplicación de la técnica
Objeto 4	E04	Descripción del programa para la aplicación de la técnica
Objeto 5	ICFB	Impreso del código fuente sin faltas de los programas (cmdline, ntree, nametbl)
Objeto 6	ICFM	Impreso del código fuente sin faltas de los programas (cmdline, ntree, nametbl)
Objeto 7	PE	Programa ejecutable
Objeto 8	Tabla de diseño	Tabla de diseño experimental

3.2. Modelado del Dominio

Para modelar el dominio de la línea de producto para experimentación se utilizó el Método FODA y se implementó en la herramienta FeatureIDE.

3.2.1. Modelo Genérico

El modelo genérico de una LPS para experimentación se implementó basado en la matriz de la **Tabla 5** a partir de la cual se obtuvieron las características (features) del proceso experimental.

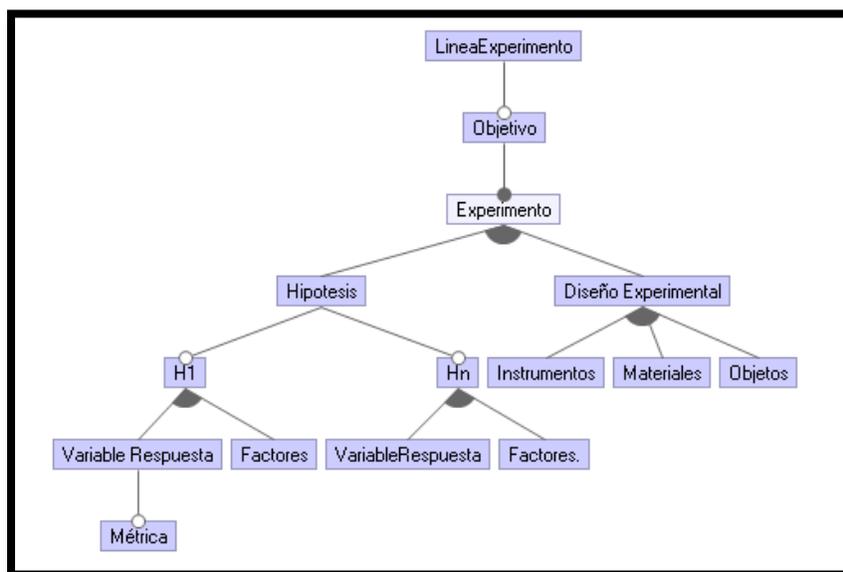


Figura 3. Modelo de Dominio para Experimentación

3.2.2. Modelo Técnicas de Evaluación de Código

El modelo genérico junto con el análisis de contexto sirve como punto de partida para estructurar el modelo de dominio del experimento Técnicas de Evaluación de Código.

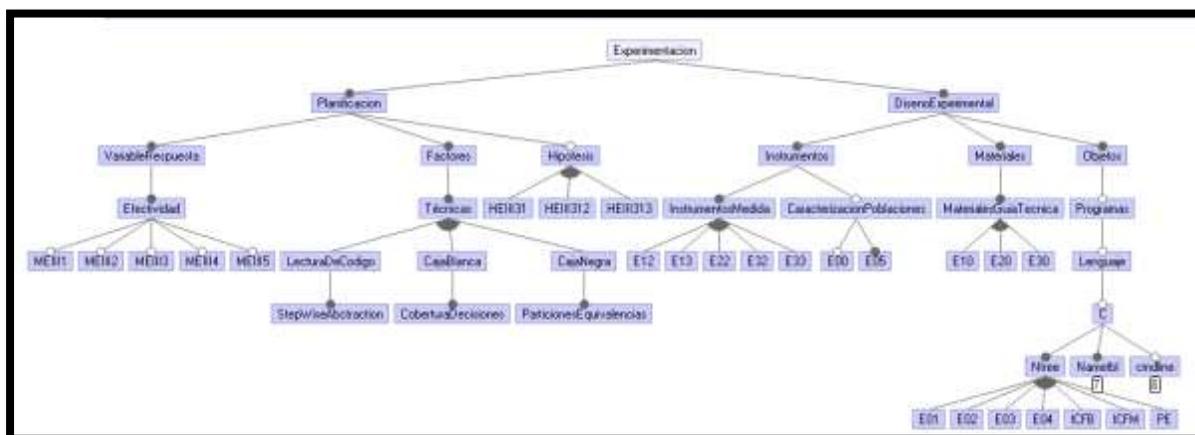


Figura 4. Modelo de Dominio Técnicas de Evaluación de Código

Para realizar la configuración de la LPS es necesario definir un conjunto de reglas que permitan la relación entre característica del modelo.

Por ejemplo, en el experimento de Técnicas de Evaluación de código si selecciono la hipótesis HEIII312 se debe relacionar con la métrica MEIII1, MEIII3 y MEIII4. En featureIDE se denominan constraints y se implementan a través de lógica de predicados. Como se observa en la Figura 5.

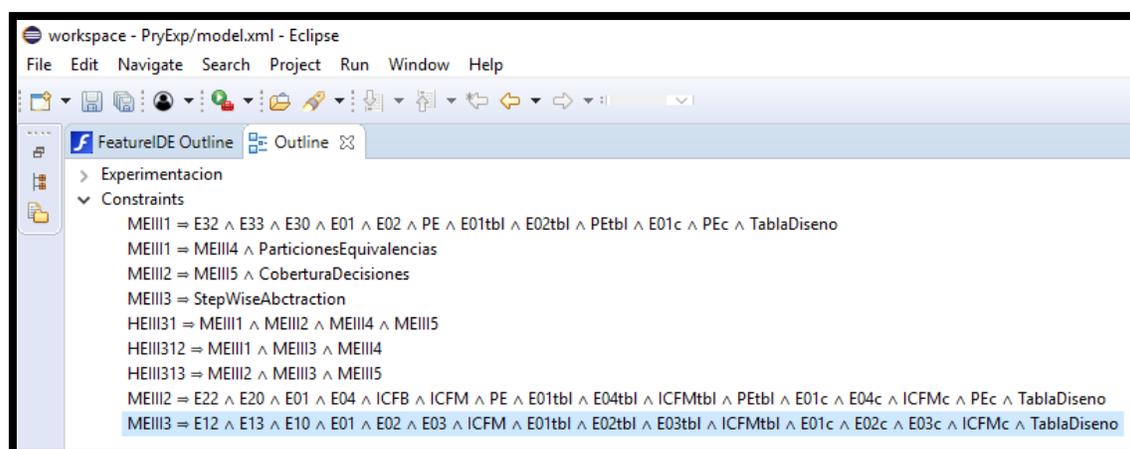


Figura 5. Reglas del Dominio

3.3. Implementación de LPS

3.3.1. Arquitectura

La arquitectura de la aplicación está conformada por dos componentes principales: una aplicación web a la cual se accede desde un ordenador mediante un navegador web, y la plataforma Firebase en la cual se gestionan los productos experimentales como se muestra en la Figura 6.

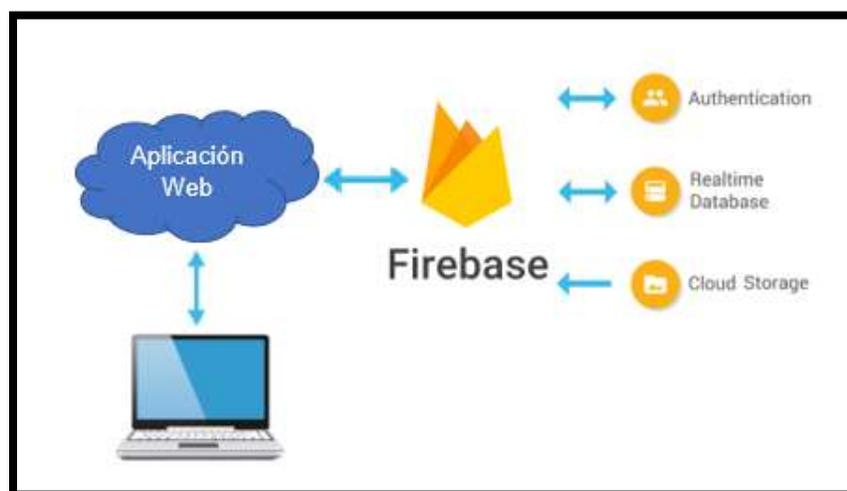


Figura 6. Arquitectura de la Aplicación

3.3.2. Configuración Firebase

En esta sección se describe la configuración del módulo de autenticación, almacenamiento (storage) y la base de Datos Realtime para el desarrollo de la aplicación.

Se puede acceder a firebase través de un navegador en <https://firebase.google.com/> como se muestra en la Figura 7.



Figura 7. Pantalla Principal de Firebase

Ingresar a la consola: <https://console.firebase.google.com/> y seleccionar Añadir Proyecto como se muestra en la Figura 8.

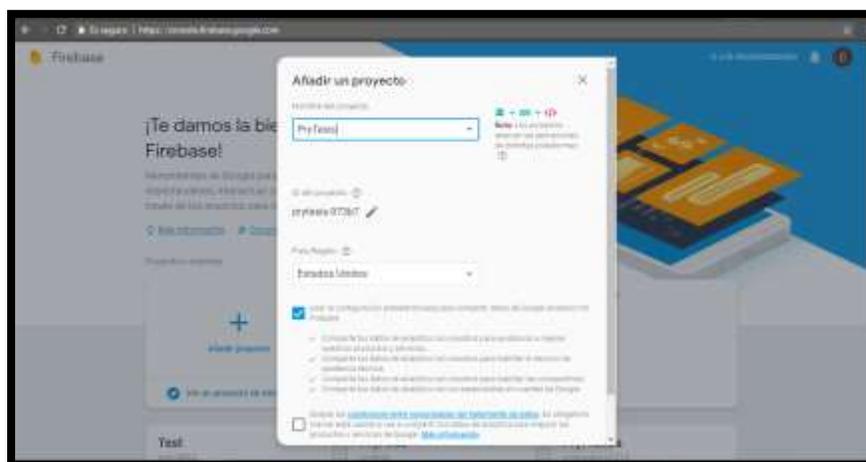


Figura 8. Consola de Firebase

Para conectar la aplicación web con Firebase se debe añadir el siguiente código, que es suministrado desde la consola de Firebase, los parámetros pueden cambiar dependiendo del proyecto.

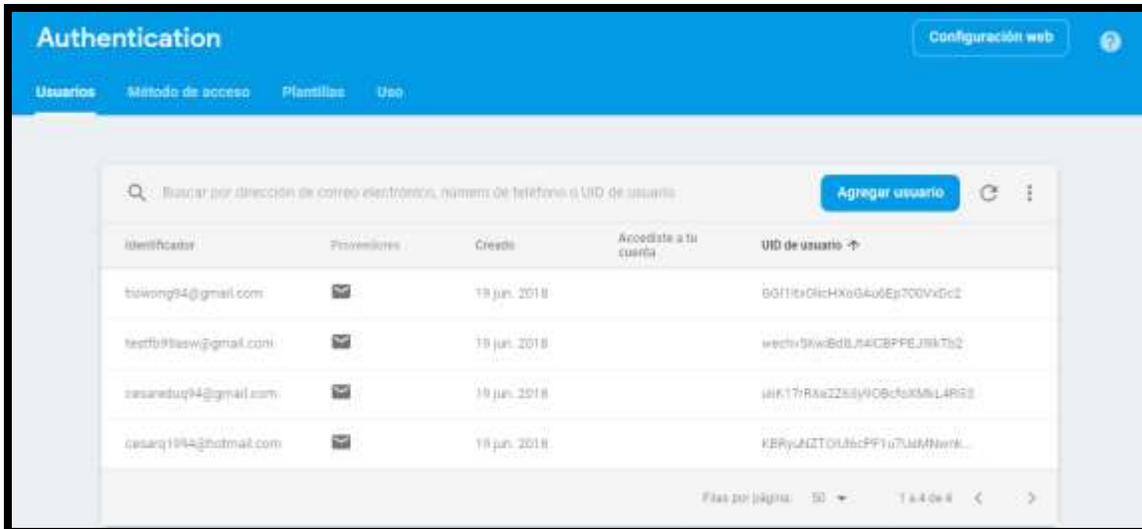
```

<script src="https://www.gstatic.com/firebasejs/5.0.4/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyC-0r0gduGSFCq9ms7Ecslv19RwKfemifM",
    authDomain: "prytesis-a0b28.firebaseio.com",
    databaseURL: "https://prytesis-a0b28.firebaseio.com",
    projectId: "prytesis-a0b28",
    storageBucket: "prytesis-a0b28.appspot.com",
    messagingSenderId: "466123016589"
  };
  firebase.initializeApp(config);
</script>

```

Figura 9. Script Firebase

Firestore permite realizar la autenticación de usuarios a través de distintos proveedores como, Facebook, Twitter, Github, email, etc. La aplicación está dirigida para experimentadores en IS, entonces se definió la creación de usuarios mediante email, como se aprecia en la Figura 10.



The screenshot shows the 'Authentication' section of the Firebase console. It features a search bar at the top with the text 'Buscar por dirección de correo electrónico, número de teléfono o UID de usuario'. Below the search bar is a table with the following columns: 'Identificador', 'Proveedores', 'Creado', 'Accede a tu cuenta', and 'UID de usuario'. The table contains four rows of user data, each with a unique UID and a creation date of 18 Jun 2018.

Identificador	Proveedores	Creado	Accede a tu cuenta	UID de usuario
tsuwong94@gmail.com	✉	18 Jun 2018		60F11xOlcHXoG4o6Ep700Vx5c2
teedb91sew@gmail.com	✉	18 Jun 2018		wectb5kwiBdLjA4CBPFEJ8kTb2
vesaneduq4@gmail.com	✉	18 Jun 2018		uK1T7RkzZ23j90BcfuXm6L4R53
qsarq1994@hotmail.com	✉	18 Jun 2018		KBRyUHZTOLmzPF1u7UMNw9k...

Figura 10. Autenticación Firebase

Para la gestión de los materiales experimentales se utilizó el servicio Cloud Storage de Firebase, el cual permite almacenar imágenes, audios, videos, documentos, en el

servidor de Google Cloud Storage, agregando seguridad al proceso de subida y descarga de archivos desde la aplicación web.

En la Figura 11 se puede observar el nombre del archivo, el tamaño, tipo o extensión del archivo y la fecha de almacenamiento.

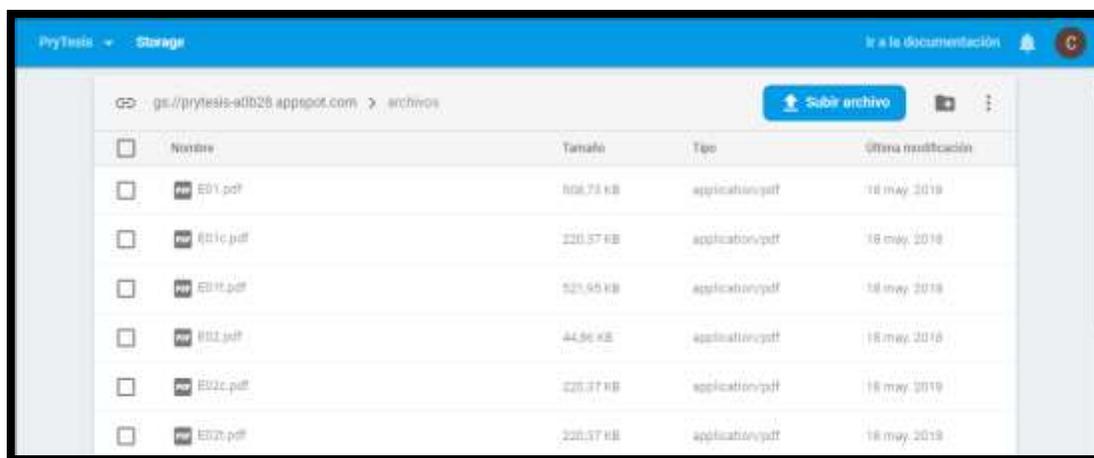


Figura 11. Almacenamiento Firebase

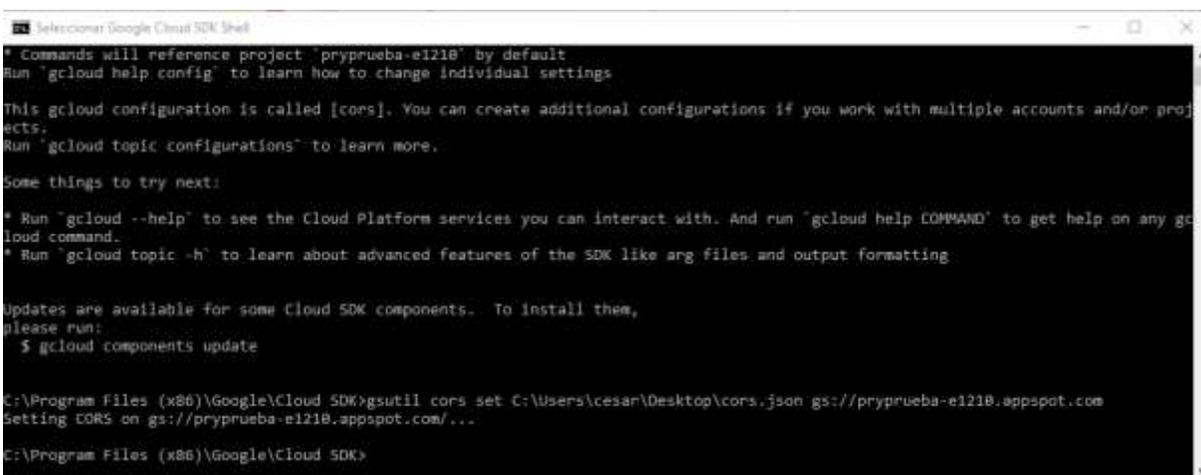
Para descargar los productos experimentales directamente en el navegador, se debe configurar el depósito de Cloud Storage para el acceso multiorigen (CORS). Esto se puede hacer con la herramienta de línea de comandos gsutil, la cual se puede instalar desde https://cloud.google.com/storage/docs/gsutil_install?hl=es-419

Se debe crear un archivo de configuración denominado cors.json con el código que se observa en la Figura 12.

```
[
  {
    "origin": ["*"],
    "method": ["GET"],
    "maxAgeSeconds": 3600
  }
]
```

Figura 12. CORS.json

Ejecutar el comando `gsutil cors set cors.json gs://<your-cloud-storage-bucket>` para implementar estas restricciones.



```
Selecionar Google Cloud SDK Shell
* Commands will reference project 'pryprueba-e1218' by default
Run 'gcloud help config' to learn how to change individual settings

This gcloud configuration is called [cors]. You can create additional configurations if you work with multiple accounts and/or projects.
Run 'gcloud topic configurations' to learn more.

Some things to try next:
* Run 'gcloud --help' to see the Cloud Platform services you can interact with. And run 'gcloud help COMMAND' to get help on any gcloud command.
* Run 'gcloud topic -h' to learn about advanced features of the SDK like arg files and output formatting

Updates are available for some Cloud SDK components. To install them, please run:
$ gcloud components update

C:\Program Files (x86)\Google\Cloud SDK>gsutil cors set C:\Users\cesar\Desktop\cors.json gs://pryprueba-e1218.appspot.com
Setting CORS on gs://pryprueba-e1218.appspot.com/...

C:\Program Files (x86)\Google\Cloud SDK>
```

Figura 13. Configurar CORS

Realtime Database utiliza una estructura tipo JSON en forma de árbol para almacenar la información.

Para gestionar los el diseño experimental se crearon los nodos Instrumentos, Materiales y Objetos, en los cuales se almacenan la línea de experimento, descripción, el nombre y la url.

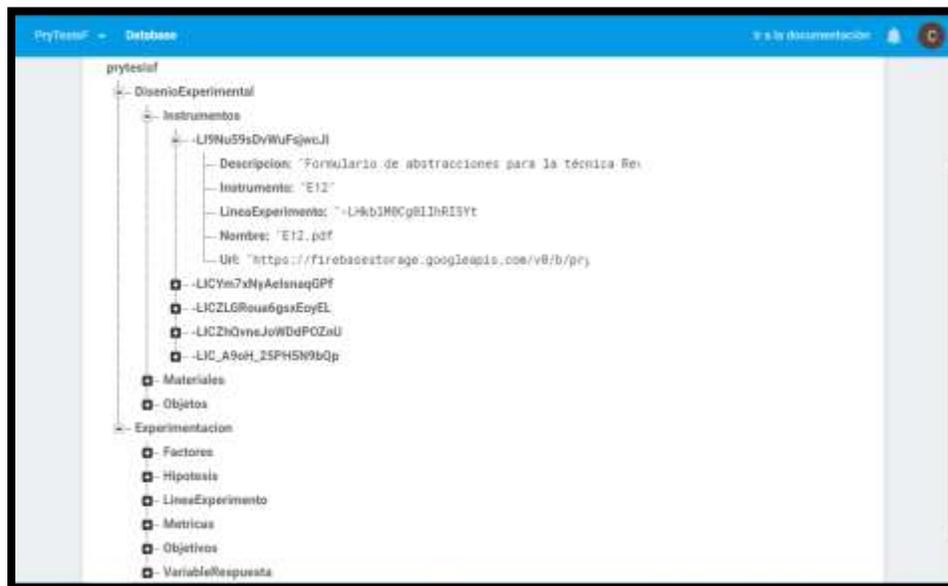


Figura 14. Realtime Database Experimentación

3.3.3. Descripción de la interfaz

Pantalla de Inicio de sesión

La pantalla muestra un formulario en el cual el usuario debe colocar su correo y contraseña para iniciar sesión y acceder a la aplicación “GESPROEXP”.



Figura 15. Pantalla de Ingreso

Pantalla de registro

Esta pantalla permita al usuario crear una cuenta, ingresando los datos de correo y contraseña correctamente en el formulario.



The screenshot shows a registration form for 'GESPROEXP'. It features three input fields: 'Correo', 'Contraseña', and 'Confirmar Contraseña'. A teal 'Regístrame' button is positioned below the fields. At the bottom, there is a link that says 'Tienes una cuenta Ingresa ahora'.

Figura 16. Pantalla de registro de usuario

Pantalla Principal

La pantalla principal (Home) muestra un menú horizontal a los cuales puede acceder el usuario, junto con la información de la aplicación web (logo y descripción).



Figura 17. Pantalla Principal

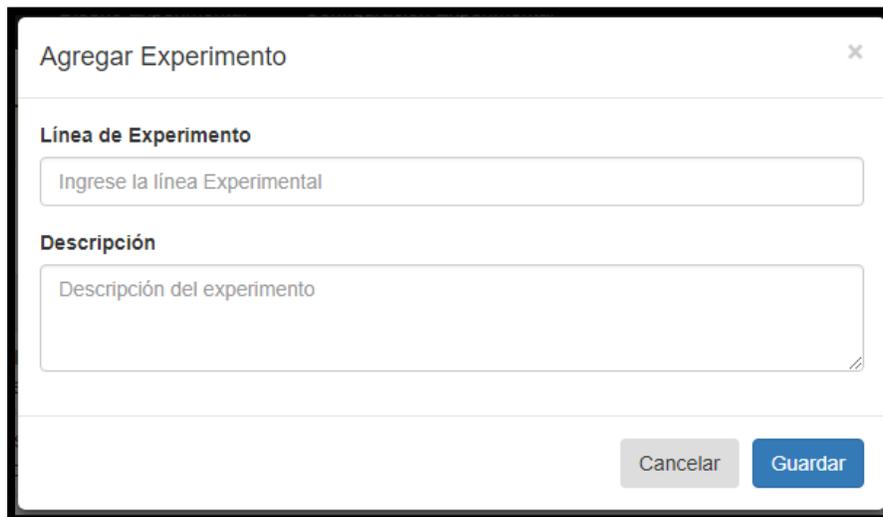
Gestión de Experimentos

Esta interfaz cuenta con un botón que permite crear un nuevo experimento, además se visualiza una tabla de los experimentos almacenados en RealTime Database de Firebase, junto con 2 botones que realizan las acciones de modificar y eliminar.



Figura 18. Gestión de Experimentos

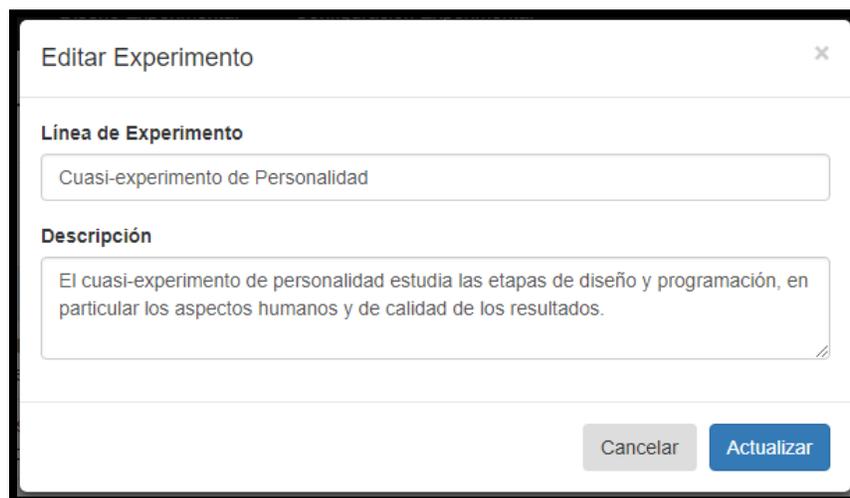
Una vez seleccionada la opción “Nuevo experimento” se despliega una ventana con un formulario “Agregar Experimento” en el cual se ingresa la línea Experimental y su descripción.



The image shows a dialog box titled "Agregar Experimento" with a close button (X) in the top right corner. It contains two input fields: "Línea de Experimento" with the placeholder text "Ingrese la línea Experimental" and "Descripción" with the placeholder text "Descripción del experimento". At the bottom right, there are two buttons: "Cancelar" (disabled) and "Guardar" (active).

Figura 19. Agregar Experimento

Si presionamos la opción Modificar se despliega una ventana en la cual se visualiza el formulario con los datos del experimento seleccionado junto a dos botones de cancelar y actualizar.



The image shows a dialog box titled "Editar Experimento" with a close button (X) in the top right corner. It contains two input fields: "Línea de Experimento" with the text "Cuasi-experimento de Personalidad" and "Descripción" with the text "El cuasi-experimento de personalidad estudia las etapas de diseño y programación, en particular los aspectos humanos y de calidad de los resultados.". At the bottom right, there are two buttons: "Cancelar" (disabled) and "Actualizar" (active).

Figura 20. Modificar Experimento

La opción eliminar muestra un mensaje de confirmación previo a borrar el experimento seleccionado.

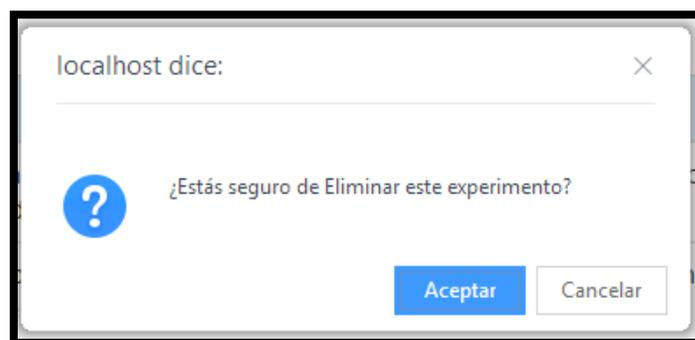


Figura 21. Eliminar Experimento

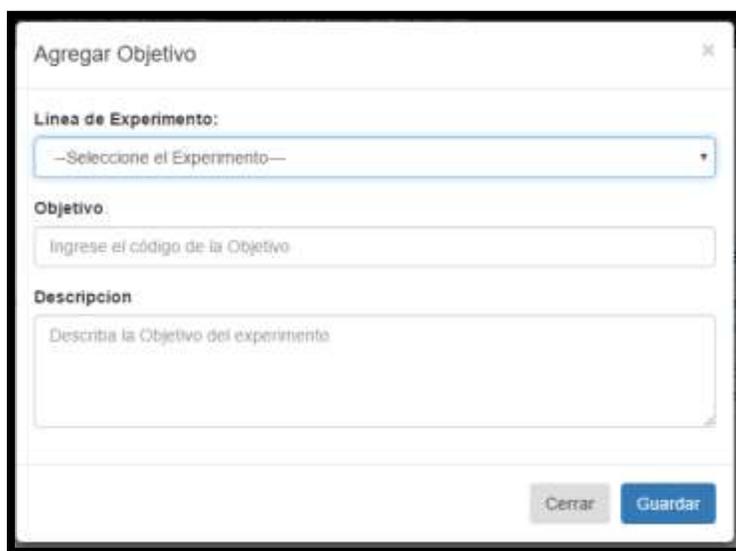
Gestión de Objetivos

Esta interfaz cuenta con un botón que permite crear un nuevo objetivo, además se visualiza una tabla con los objetivos almacenados en RealTime Database de Firebase, junto con 2 botones que realizan las acciones de modificar y eliminar.

Objetivo	Descripción	Experimento	Acciones
O1	Análisis y comparación de las relaciones existentes entre los cinco factores de la personalidad a nivel de equipo, las características de la tarea y los procesos de equipo	Cuasi-experimento de Personalidad	 
O11	Estudiar la efectividad y eficiencia de distintas técnicas de evaluación de código de código	Comparación de Técnicas de evaluación de Código	 

Figura 22. Gestionar Objetivos

Una vez seleccionada la opción “Nuevo Objetivo” se despliega una ventana con un formulario “Agregar Objetivo” en el cual se selecciona la línea Experimental a la que pertenece el objetivo y se ingresa su código y su descripción.



The image shows a web form titled "Agregar Objetivo" with a close button (X) in the top right corner. The form contains the following fields:

- Línea de Experimento:** A dropdown menu with the placeholder text "--Seleccione el Experimento--".
- Objetivo:** A text input field with the placeholder text "Ingrese el código de la Objetivo".
- Descripcion:** A larger text area with the placeholder text "Describa la Objetivo del experimento".

At the bottom right of the form, there are two buttons: "Cerrar" (grey) and "Guardar" (blue).

Figura 23. Agregar Objetivo

Si presionamos la opción Modificar se despliega una ventana en la cual se visualiza el formulario con los datos del objetivo seleccionado junto a dos botones de cancelar y actualizar.

Editar Objetivo

Linea de Experimento:
Cuasi-experimento de Personalidad

Objetivo
O1

Descripcion
Analizar y comparar las relaciones existentes entre los cinco factores de la personalidad a nivel de equipo, las características de la tarea y los procesos de equipo

Cerrar Actualizar

Figura 24. Modificar Objetivo

GESTIÓN DE HIPÓTESIS

Esta interfaz cuenta con un botón que permite crear una nueva hipótesis, además se visualiza una tabla con las hipótesis almacenados en RealTime Database de Firebase, junto con 2 botones que realizan las acciones de modificar y eliminar.

Hipótesis

Nueva Hipótesis

Hipótesis	Descripción	Experimento	Acciones
H010-02	No existe relación entre las características de la tarea	Cuasi-experimento de Personalidad	
H010-03	No existe relación entre los procesos de equipo	Cuasi-experimento de Personalidad	
HE1030	No hay diferencia en la efectividad entre las técnicas de partición de clases de equivalencia, cobertura de sentencias	Comparación de Técnicas de evaluación de Código	
HE1031	Hay diferencia en la efectividad entre las técnicas de partición de clases de equivalencia, cobertura de decisión	Comparación de Técnicas de evaluación de Código	

Figura 25. Gestionar Hipótesis

Una vez seleccionada la opción “Nueva Hipótesis” se despliega una ventana con un formulario “Agregar Hipótesis” en el cual se selecciona la línea Experimental a la que pertenece la hipótesis, se ingresa su código y su descripción.

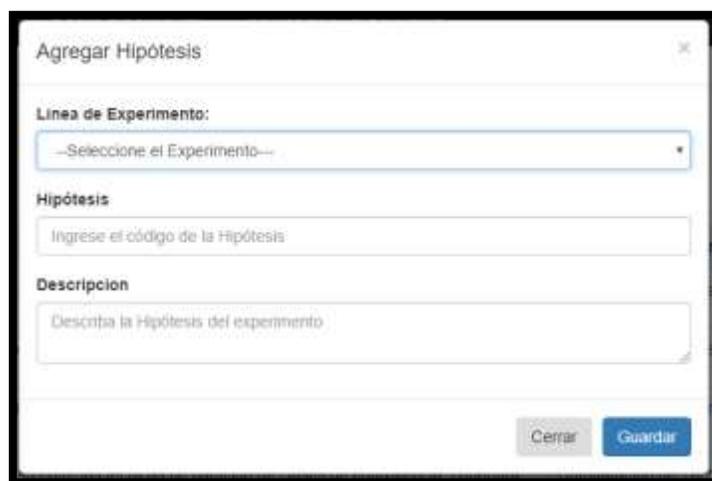


Figura 26. Agregar Hipótesis

Si presionamos la opción Modificar se despliega una ventana en la cual se visualiza el formulario con los datos de la hipótesis seleccionada junto a dos botones de cancelar y actualizar.

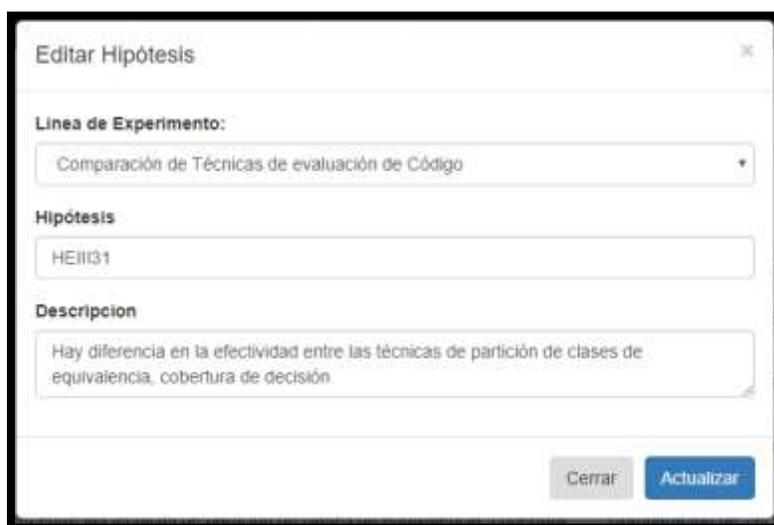


Figura 27. Modificar Hipótesis

GESTIÓN DE VARIABLE RESPUESTA

Esta interfaz cuenta con un botón que permite crear una nueva variable respuesta, además se visualiza una tabla con las variables respuestas previamente almacenadas en RealTime Database de Firebase, junto con 2 botones que realizan las acciones de modificar y eliminar.



Variable Respuesta	Experimento	Acciones	
Satisfacción del trabajo realizado	Cuasi-experimento de Personalidad		
Calidad del Software Desarrollado	Cuasi-experimento de Personalidad		
Visibilidad	Comparación de Técnicas de evaluación de Código		
Efectividad	Comparación de Técnicas de evaluación de Código		

Figura 28. Gestionar Variable Respuesta

Una vez seleccionada la opción “Nueva Variable Respuesta” se despliega una ventana con un formulario “Agregar Variable Respuesta” en el cual se selecciona la línea Experimental a la que pertenece y se ingresa la variable respuesta.



Agregar Variable Respuesta

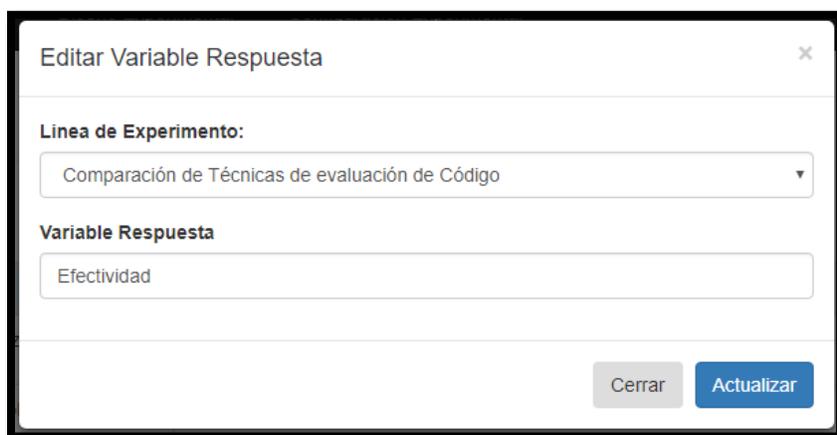
Linea de Experimento:
--Seleccione el Experimento--

Variable Respuesta
Ingrese Variable Respuesta

Cerrar Guardar

Figura 29. Agregar Variable Respuesta

Si presionamos la opción Modificar se despliega una ventana en la cual se visualiza el formulario con los datos de la variable respuesta seleccionada junto a dos botones de cancelar y actualizar.



Editar Variable Respuesta

Linea de Experimento:
Comparación de Técnicas de evaluación de Código

Variable Respuesta
Efectividad

Cerrar Actualizar

Figura 30. Modificar Variable Respuesta

GESTIÓN DE MÉTRICA

Esta interfaz cuenta con un botón que permite crear una nueva métrica, además se visualiza una tabla con las métricas previamente almacenadas en RealTime Database de Firebase, junto con 2 botones que realizan las acciones de modificar y eliminar.

Métrica	Descripción	Acciones
MEI01	Percentage of F1, F2 and F3 for which a subject generates a test case that exercises the faults	[Edit] [Delete] [Download]
MEI02	Percentage of F4, F5 and F6 for which a subject generates a test case that exercises the faults	[Edit] [Delete] [Download]
MEI03	Percentage of F1, F2, F3, F4, F5 and F6 reported by subjects	[Edit] [Delete] [Download]
MEI04	Percentage of F4, F5 and F6 for which a subject generates a test case that exercises the faults	[Edit] [Delete] [Download]
MEI05	Percentage of F1, F2 and F3 for which a subject generates a test case that exercises the faults	[Edit] [Delete] [Download]
M1	Número de módulos de acoplamiento	[Edit] [Delete] [Download]
M2	Número de defectos detectados por el conjunto de casos de prueba ejecutado automáticamente	[Edit] [Delete] [Download]

Figura 31. Gestionar Métrica

Una vez seleccionada la opción “Nueva Métrica” se despliega una ventana con un formulario en el cual se selecciona la línea experimental, se ingresa el código, la descripción y el archivo perteneciente a la métrica.

Agregar Métricas

Línea de Experimento:
 --Seleccione el Experimento--

Métrica
 Ingrese la Métrica

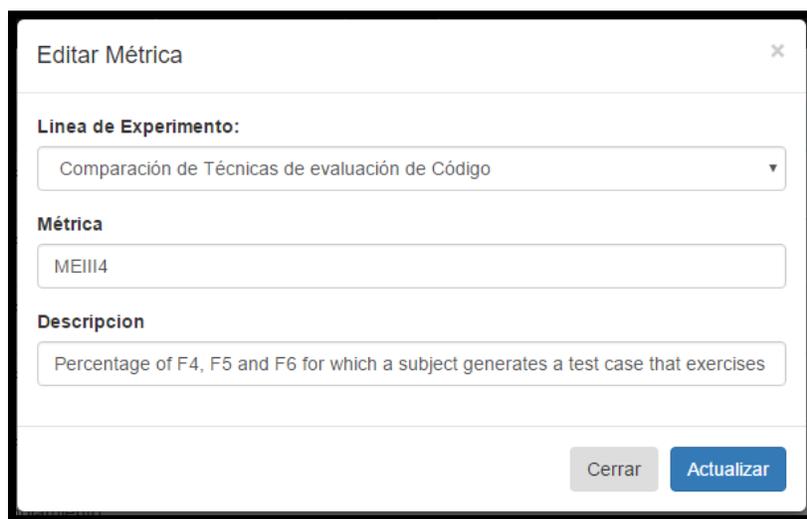
Descripción
 Describa la Métrica

Archivo:
 Seleccionar archivos... **Examinar ...**

Cerrar **Guardar**

Figura 32. Agregar Métrica

Si presionamos la opción Modificar se despliega una ventana en la cual se visualiza el formulario con los datos de la métrica seleccionada junto a dos botones de cancelar y actualizar.

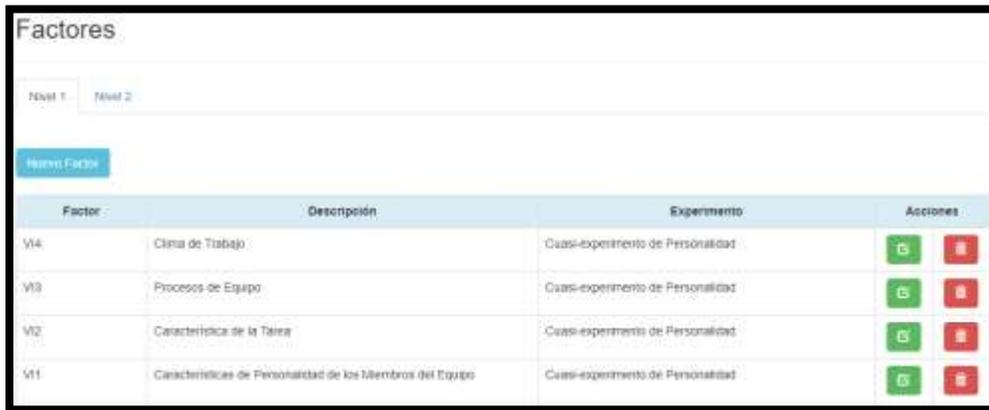


The image shows a modal window titled "Editar Métrica" with a close button (x) in the top right corner. The form contains three sections: "Linea de Experimento:" with a dropdown menu showing "Comparación de Técnicas de evaluación de Código"; "Métrica" with a text input field containing "MEIII4"; and "Descripcion" with a text input field containing "Percentage of F4, F5 and F6 for which a subject generates a test case that exercises". At the bottom right, there are two buttons: "Cerrar" (grey) and "Actualizar" (blue).

Figura 33. Modificar Métrica

GESTIÓN DE FACTORES

Esta interfaz cuenta con un botón que permite crear dos niveles de Factores, además se visualiza una tabla con los factores previamente almacenados en RealTime Database de Firebase, junto con 2 botones que realizan las acciones de modificar y eliminar.



Factor	Descripción	Experimento	Acciones
V4	Clima de Trabajo	Cuasi-experimento de Personalidad	 
V3	Procesos de Equipo	Cuasi-experimento de Personalidad	 
V2	Característica de la Tarea	Cuasi-experimento de Personalidad	 
V1	Características de Personalidad de los Miembros del Equipo	Cuasi-experimento de Personalidad	 

Figura 34. Gestionar Factores

Una vez seleccionada la opción “Nuevo Factor” se despliega una ventana con un formulario de Agregar Factor en el cual se selecciona la línea Experimental a la que pertenece y se ingresa su código y descripción.



Agregar Factor

Línea de Experimento:
 --Seleccione el Experimento--

Factor
 Ingrese el código del Factor

Descripción
 Describa el Factor del experimento

Cancelar Guardar

Figura 35. Agregar Factor

Si presionamos la opción Modificar se despliega una ventana en la cual se visualiza el formulario con los datos de la métrica seleccionada junto a dos botones de cancelar y actualizar.

Editar Factor

Linea de Experimento:
Cuasi-experimento de Personalidad

Factor
VI1

Descripcion
Características de Personalidad de los Miembros del Equipo

Cancelar Actualizar

Figura 36. Modificar Factores

GESTIÓN DE INSTRUMENTOS

Esta interfaz cuenta con un botón que permite crear un nuevo instrumento, además se visualiza una tabla con los instrumentos previamente almacenados en RealTime Database de Firebase, junto con 3 botones que realizan las acciones de modificar, eliminar y descargar.

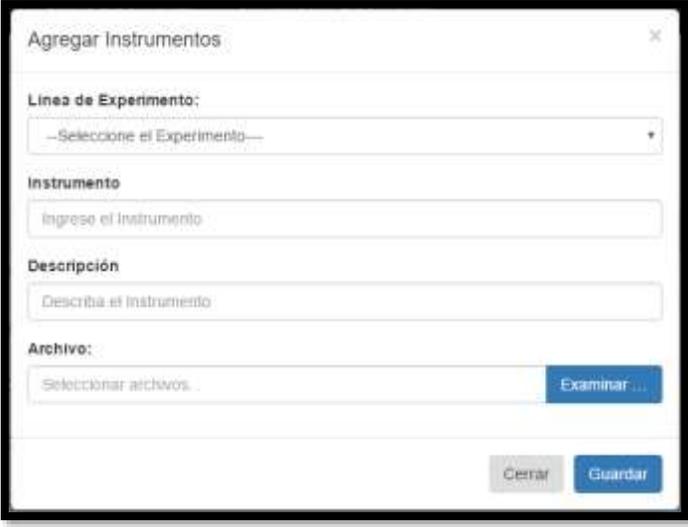
Instrumentos

Nuevo Instrumento

Instrumento	Descripción	Acciones
E12	Formulario de abstracciones para la técnica Revisión de Código	
E13	Formulario de inconsistencias para la Revisión de Código	
E33	Formulario de Datos de Prueba para las Pruebas Funcionales	

Figura 37. Gestionar Instrumentos

Una vez seleccionada la opción “Nuevo Instrumento” se despliega una ventana con un formulario en el cual se selecciona la línea experimental, se ingresa el código, la descripción y el archivo perteneciente al instrumento experimental.



The screenshot shows a web form titled "Agregar Instrumentos" with a close button (X) in the top right corner. The form contains the following fields and controls:

- Línea de Experimento:** A dropdown menu with the placeholder text "--Seleccione el Experimento--".
- Instrumento:** A text input field with the placeholder text "Ingreso el Instrumento".
- Descripción:** A text input field with the placeholder text "Describe el Instrumento".
- Archivo:** A text input field with the placeholder text "Seleccionar archivos...". To its right is a blue button labeled "Examinar...".
- At the bottom right of the form are two buttons: a grey "Cerrar" button and a blue "Guardar" button.

Figura 38. Agregar Instrumentos.

Si presionamos la opción Modificar se despliega una ventana en la cual se visualiza el formulario con los datos del instrumento seleccionado junto a dos botones de cancelar y actualizar.



The screenshot shows a web form titled "Editar Instrumento" with a close button (X) in the top right corner. The form contains the following fields and controls:

- Línea de Experimento:** A dropdown menu with the selected value "Comparación de Técnicas de evaluación de Código".
- Instrumento:** A text input field with the value "E33".
- Descripción:** A text input field with the value "Formulario de Datos de Prueba para las Pruebas Funcionales".
- At the bottom right of the form are two buttons: a grey "Cerrar" button and a blue "Actualizar" button.

Figura 39. Modificar Instrumento

GESTIÓN DE MATERIALES

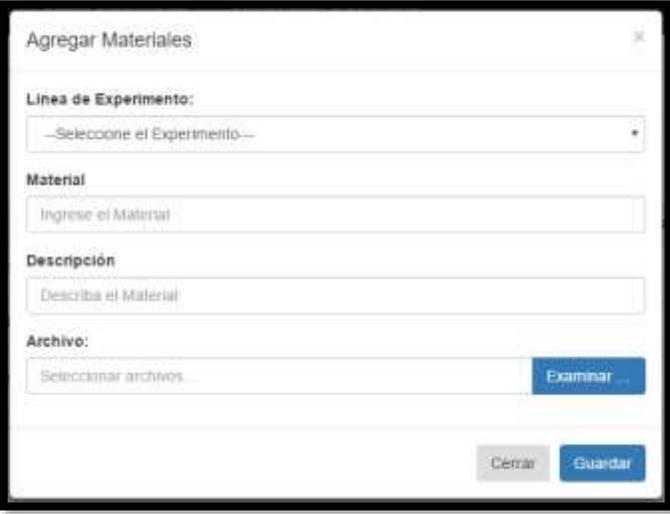
Esta interfaz cuenta con un botón que permite crear un nuevo material experimental, además se visualiza una tabla con las variables respuestas previamente almacenadas en RealTime Database de Firebase, junto con 3 botones que realizan las acciones de modificar, eliminar y descargar.



Material	Descripción	Acciones
E10	Instrucciones para aplicar la técnica Revisión de Código	  
E20	Instrucciones para aplicar la técnica Pruebas Estructurales	  
E30	Instrucciones para aplicar la técnica de Pruebas Funcionales	  

Figura 40. Gestionar Materiales

Una vez seleccionada la opción “Nuevo Material” se despliega una ventana con un formulario en el cual se selecciona la línea experimental, se ingresa el código, la descripción y el archivo perteneciente al material experimental.



Agregar Materiales

Línea de Experimento:
--Seleccione el Experimento--

Material
Ingrese el Material

Descripción
Describe el Material

Archivo:
Seleccionar archivos Examinar

Cerrar Guardar

Figura 41. Agregar Materiales

Si presionamos la opción Modificar se despliega una ventana en la cual se visualiza el formulario con los datos del material experimental seleccionado, junto a dos botones de cancelar y actualizar.



Figura 42. Modificar Material

GESTIÓN DE OBJETOS

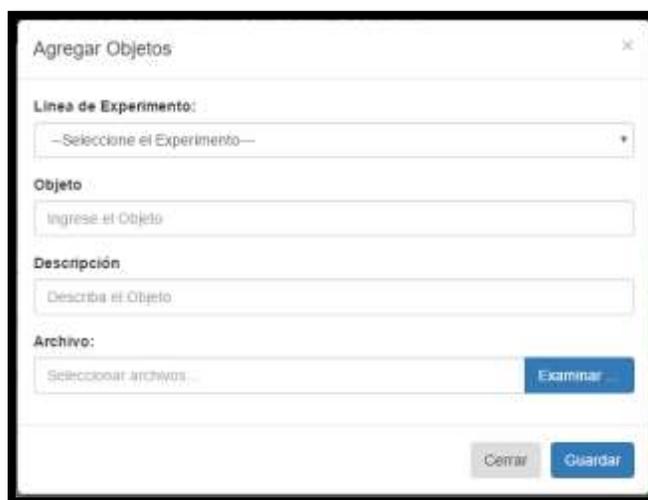
Esta interfaz cuenta con un botón que permite crear un nuevo objeto, además se visualiza una tabla con los objetos previamente almacenadas en RealTime Database de Firebase, junto con 3 botones que realizan las acciones de modificar, eliminar y descargar.



Objeto	Descripción	Acciones
Ed@ntree	Programa ejecutable12	[Edit] [Delete] [Download]
Ed@ntree	Descripción del programa para la aplicación de la técnica	[Edit] [Delete] [Download]
Ed@ntree	Descripción del programa para la aplicación de la técnica	[Edit] [Delete] [Download]
ICF@ntree	Impresión del código fuente con faltas	[Edit] [Delete] [Download]
pe	Descripción del programa para la aplicación de la técnica	[Edit] [Delete] [Download]
ICF@ntree	Impresión del código sin faltas	[Edit] [Delete] [Download]

Figura 43. Gestionar Objetos

Una vez seleccionada la opción “Nuevo Objeto” se despliega una ventana con un formulario en el cual se selecciona la línea experimental, se ingresa el código, la descripción y el archivo perteneciente al objeto experimental.



The screenshot shows a web form titled "Agregar Objetos" with a close button (X) in the top right corner. The form contains the following fields and buttons:

- Línea de Experimento:** A dropdown menu with the placeholder text "—Seleccione el Experimento—".
- Objeto:** A text input field with the placeholder text "Ingrese el Objeto".
- Descripción:** A text input field with the placeholder text "Describe el Objeto".
- Archivo:** A file selection button with the text "Seleccionar archivos..." and a blue "Examinar" button to its right.
- At the bottom right, there are two buttons: a grey "Cerrar" button and a blue "Guardar" button.

Figura 44. Agregar Objetos

Si presionamos la opción Modificar se despliega una ventana en la cual se visualiza el formulario con los datos del objeto experimental seleccionado junto a dos botones de cancelar y actualizar.



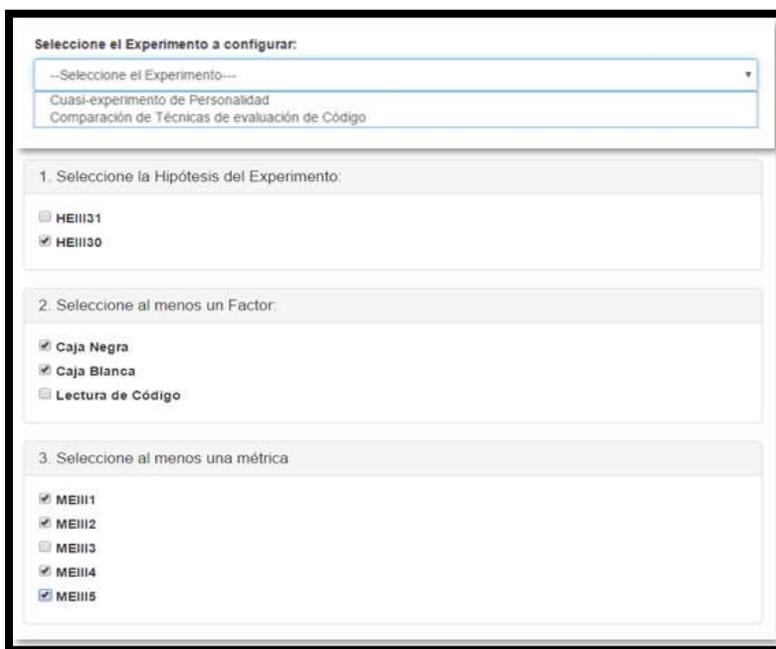
The screenshot shows a web form titled "Editar Objeto" with a close button (X) in the top right corner. The form contains the following fields and buttons:

- Línea de Experimento:** A dropdown menu with the selected value "Comparación de Técnicas de evaluación de Código".
- Objeto:** A text input field containing the value "ICFMntree".
- Descripción:** A text input field containing the value "Impresión del código fuente con faltas".
- At the bottom right, there are two buttons: a grey "Cerrar" button and a blue "Actualizar" button.

Figura 45. Modificar Objetos

CONFIGURAR EXPERIMENTO

Para obtener los productos experimentales para realizar una replicación seleccionamos la opción “Configuración Experimental”. La aplicación permite relacionar en base a la selección de la Hipótesis, todos los materiales que se requieren para instanciar la replicación de experimento



The screenshot displays a web interface for configuring an experiment. It is titled "Seleccione el Experimento a configurar:" and contains three main sections:

- Experiment Selection:** A dropdown menu with the placeholder "--Seleccione el Experimento--". Below it, two options are listed: "Cuasi-experimento de Personalidad" and "Comparación de Técnicas de evaluación de Código".
- Hypothesis Selection:** Titled "1. Seleccione la Hipótesis del Experimento:". It contains two radio button options: "HEIII31" (unchecked) and "HEIII30" (checked).
- Factor Selection:** Titled "2. Seleccione al menos un Factor:". It contains three checkbox options: "Caja Negra" (checked), "Caja Blanca" (checked), and "Lectura de Código" (unchecked).
- Metric Selection:** Titled "3. Seleccione al menos una métrica". It contains five checkbox options: "MEIII1" (checked), "MEIII2" (checked), "MEIII3" (unchecked), "MEIII4" (checked), and "MEIII5" (checked).

Figura 46. Configurar Experimento

La aplicación genera un archivo empaquetado que es descargado al sitio de almacenamiento del investigador interesado en replicar el experimento. Éste archivo empaquetado contiene todos los materiales que se utilizarán en la replicación.

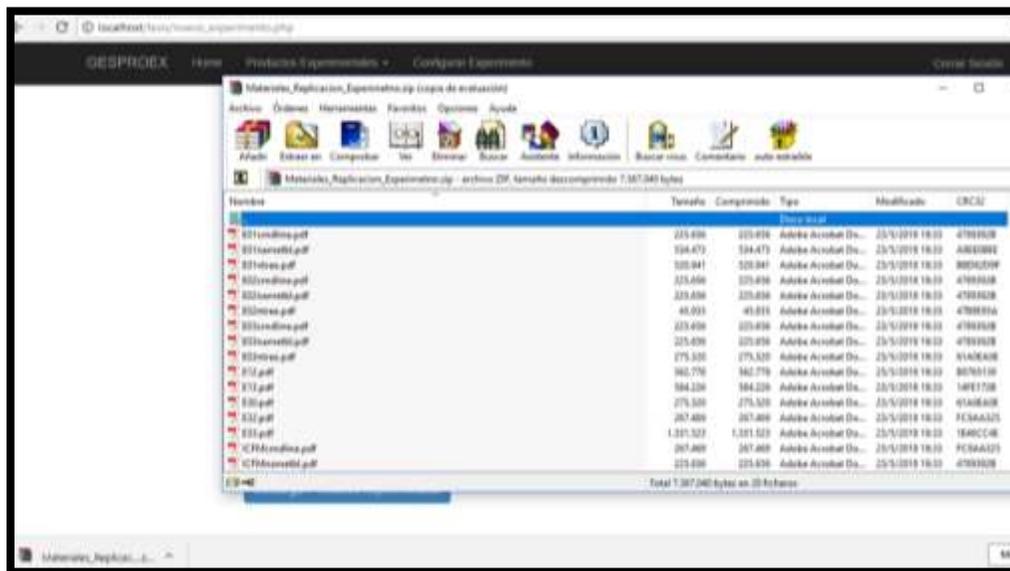


Figura 47. Productos Experimentales

3.4. Limitaciones

- La aplicación web “GESTPROEX” se centra en la gestión de productos experimentales de las fases de definición y diseño de un experimento.
- La aplicación web solo se ejecutará en Navegadores Web de ordenares PC.
- El servicio Storage tiene una capacidad máxima de 5 Gb de almacenamiento para guardar los productos experimentales.
- La base de datos Realtime permite hasta 100 conexiones simultáneas mediante la plicación web y la capacidad para almacenar datos de la línea de producto experimental es de 1 Gb.

3.5. Despliegue

Se utilizará la plataforma Heroku para alojar y desplegar la aplicación web, ya que permite agilizar el proceso de publicar nuestras aplicaciones sin tener que preocuparnos demasiado en configurar nuestros propios servidores.

Ingresar a <https://dashboard.heroku.com/apps> y crear una nueva aplicación.

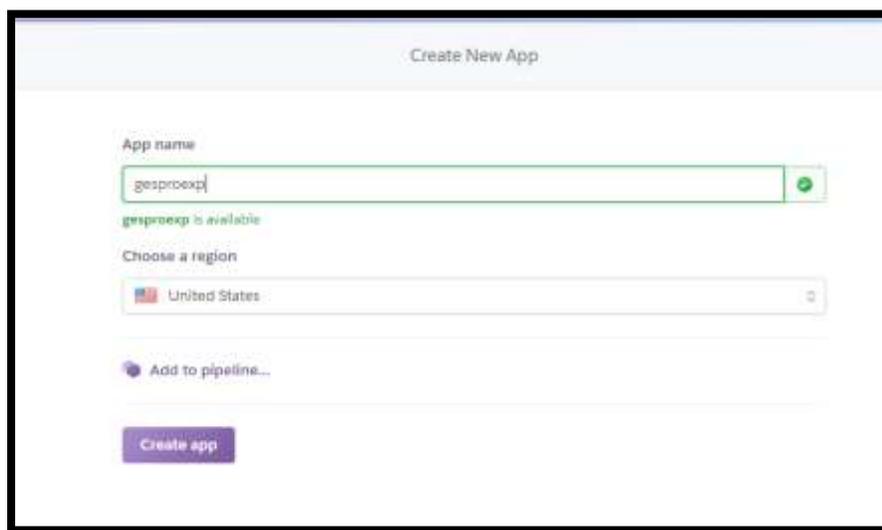


Figura 48. Crear App Heroku

En el panel de administración de Heroku, seleccionar Heroki Git como método de despliegue.

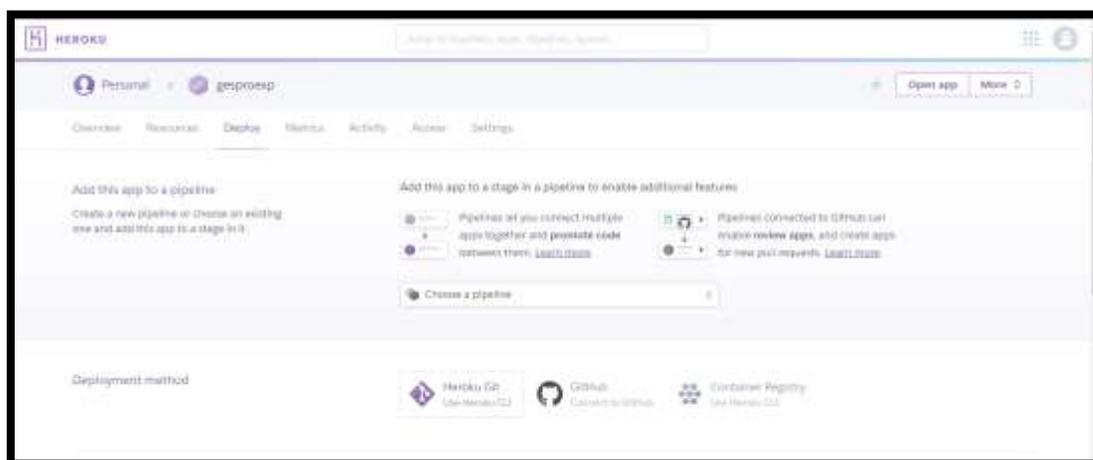
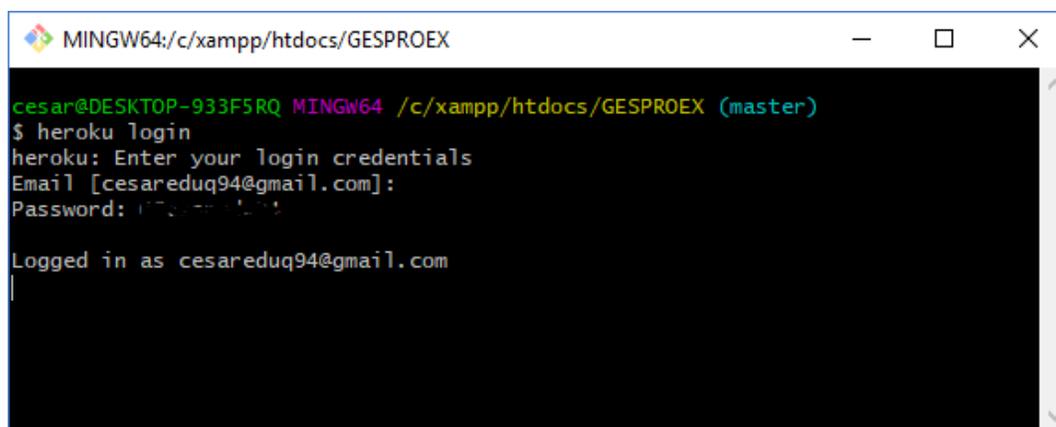


Figura 49. Panel de Administración Heroku

Descargar e instalar HerokuCLI desde <https://devcenter.heroku.com/articles/heroku-cli>. Una vez instalado ingresar a la consola de comandos e iniciar sesión con el comando \$ heroku init.

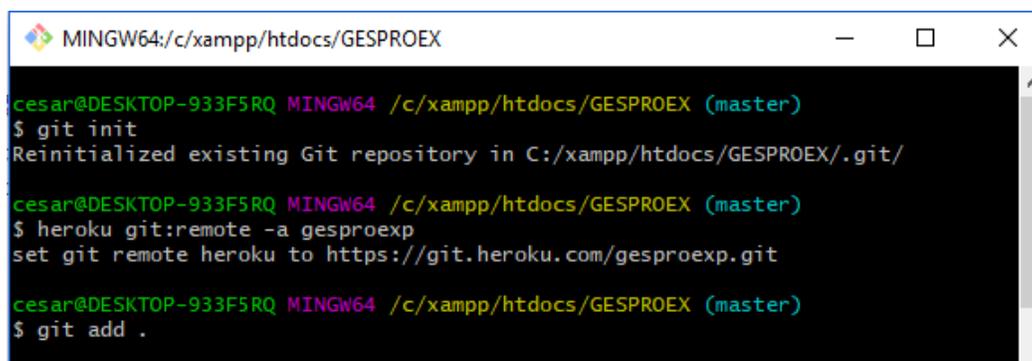
A screenshot of a Windows command prompt window titled 'MINGW64:/c/xampp/htdocs/GESPROEX'. The prompt shows the user 'cesar@DESKTOP-933F5RQ' in the 'MINGW64' environment at the directory '/c/xampp/htdocs/GESPROEX (master)'. The user enters the command '\$ heroku login'. The terminal displays the Heroku login process: 'heroku: Enter your login credentials', 'Email [cesareduq94@gmail.com]:', 'Password: [REDACTED]', and finally 'Logged in as cesareduq94@gmail.com'.**Figura 50.** Iniciar sesión en Heroku

Iniciar un repositorio git en el directorio en el cual se encuentra el código de la aplicación web con los siguientes comandos:

```
$ cd C:\xampp\htdocs\GESPROEX
```

```
$ git init
```

```
$ Heroku git:remote -a gesproexp
```

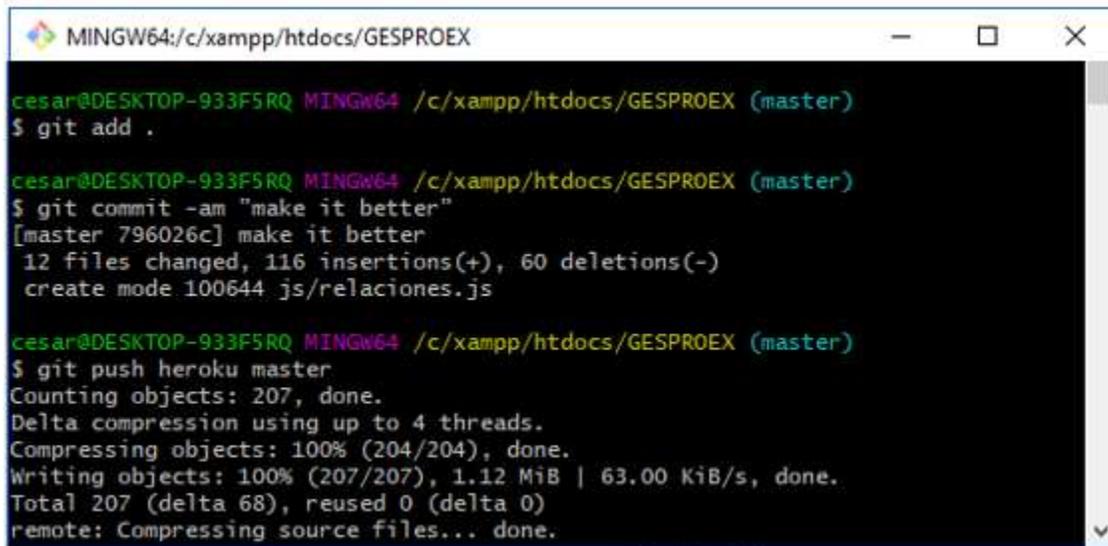
A screenshot of a Windows command prompt window titled 'MINGW64:/c/xampp/htdocs/GESPROEX'. The prompt shows the user 'cesar@DESKTOP-933F5RQ' in the 'MINGW64' environment at the directory '/c/xampp/htdocs/GESPROEX (master)'. The user enters the command '\$ git init', and the terminal displays 'Reinitialized existing Git repository in C:/xampp/htdocs/GESPROEX/.git/'. The user then enters '\$ heroku git:remote -a gesproexp', and the terminal displays 'set git remote heroku to https://git.heroku.com/gesproexp.git'. Finally, the user enters '\$ git add .'.**Figura 51.** Iniciar Repositorio Git

Finalmente agregar el código fuente al repositorio y desplegarlo en Heroku.

```
$ git add .
```

```
$ git commit -am "make it better"
```

```
$ git push heroku master
```



```
MINGW64:/c:/xampp/htdocs/GESPROEX
cesar@DESKTOP-933F5RQ MINGW64 /c:/xampp/htdocs/GESPROEX (master)
$ git add .

cesar@DESKTOP-933F5RQ MINGW64 /c:/xampp/htdocs/GESPROEX (master)
$ git commit -am "make it better"
[master 796026c] make it better
12 files changed, 116 insertions(+), 60 deletions(-)
create mode 100644 js/relaciones.js

cesar@DESKTOP-933F5RQ MINGW64 /c:/xampp/htdocs/GESPROEX (master)
$ git push heroku master
Counting objects: 207, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (204/204), done.
Writing objects: 100% (207/207), 1.12 MiB | 63.00 KiB/s, done.
Total 207 (delta 68), reused 0 (delta 0)
remote: Compressing source files... done.
```

Figura 52. Despliegue en Heroku

CAPÍTULO IV

PRUEBAS DE LA APLICACIÓN

4.1. Introducción

Una vez concluido el desarrollo de la aplicación web, se realiza un plan de pruebas para evaluar su funcionalidad. Las pruebas que se ejecutará para realizar la validación es la Prueba de instanciación

4.2. Prueba de instanciación

La instanciación se la realiza de acuerdo a la hipótesis experimental, ésta puede ser nula o afirmativa. En este caso se evaluarán las Hipótesis HEIII31, HEIII312 Y HEIII313 las cuales se describen a continuación:

Tabla 7

Hipótesis Técnicas de Evaluación de código

Hipótesis	Descripción
HEIII31	Hay diferencia en la efectividad entre las técnicas de partición de clases de equivalencia, cobertura de decisión
HEIII312	Hay diferencia en la efectividad entre las técnicas de partición de clases de equivalencia y lectura de código
HEIII313	Hay diferencia en la efectividad entre las técnicas de cobertura de decisión y lectura de código

Las pruebas se realizarán acorde a los productos experimentales generados en el módulo de Configuración Experimental de la aplicación web “GESPROEXP”.

4.2.1. Caso de Prueba HEIII31

Tabla 8

Prueba de Instanciación HEIII31

Experimento: Técnicas de Evaluación de Código		
Hipótesis: HEIII31		
	Criterio	Resultado
Productos Experimentales	E22	Al realizar la configuración del experimento técnicas de evaluación de código, instanciando la hipótesis HEIII31 se obtienen los productos experimentales que se muestra en la figura Figura 53
	E32	
	E33	
	E20	
	E30	
	E01ntree	
	E02ntree	
	E04ntree	
	ICFBntree	
	ICFMntree	
	PEntree	
	E01nametbl	
	E02nametbl	
	E04nametbl	
	ICFMnametbl	
	PEnametbl	
	E01cmdline	
	E02cmdline	
	E04cmdline	
	ICFMcmdline	
PEcmdline		
Tabla de Diseño		

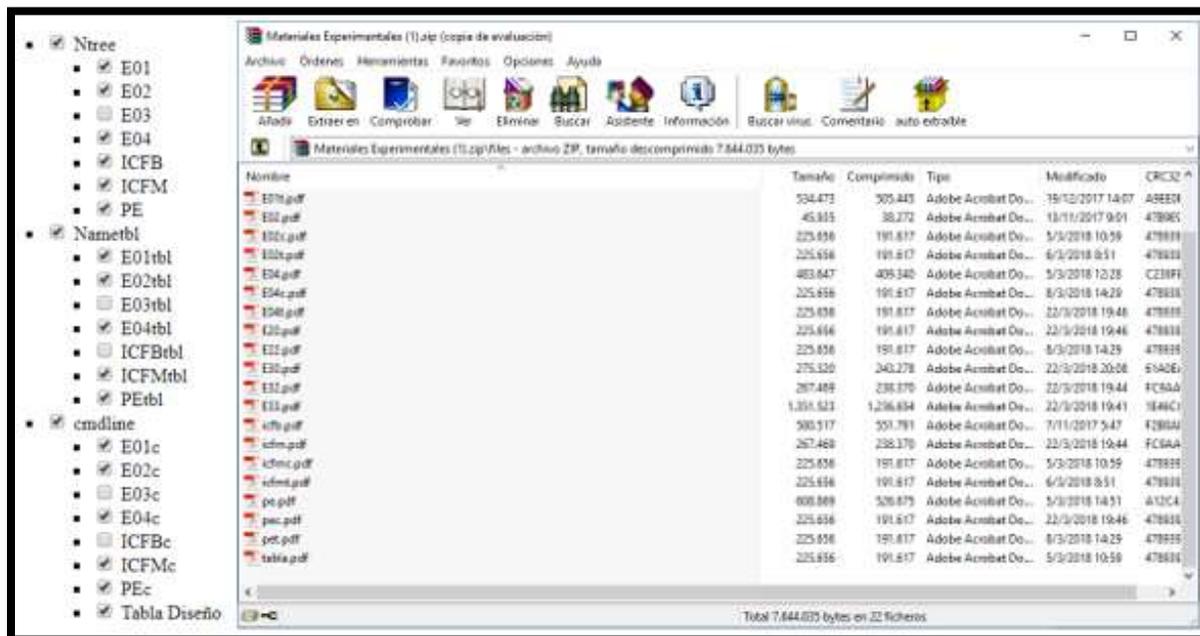


Figura 53. Instanciación HEIII31

En la tabla 8 establece los parámetros necesarios para ejecutar la hipótesis HEIII31, en la cual se puede visualizar la lista de materiales experimentales que se utilizan para la replicación de este experimento.

En la figura 53 se puede observar que la lista de materiales descargados por la aplicación “GESPROEXP” es igual a la lista que proporcionó el experto en replicación.

4.2.2. Caso de Prueba HEIII312

Tabla 9*Prueba de instanciación HEIII312*

Experimento: Técnicas de Evaluación de Código		
Hipótesis: HEIII312		
	Criterio	Resultado
Productos Experimentales	E12	Observar la Figura 54
	E13	
	E32	
	E33	
	E10	
	E30	
	E01ntree	
	E02ntree	
	E03ntree	
	ICFMntree	
	PEntree	
	E01nametbl	
	E02nametbl	
	E03nametbl	
	ICFMnametbl	
	PEnametbl	
	E01cmdline	
	E02cmdline	
	E03cmdline	
	ICFMcmdline	
	PEcmdline	
Tabla de diseño		

Nombre	Tamaño	Comprimido	Tipo	Modificado	CRC32
Doces local					
E01.pdf	526.941	484.278	Adobe Acrobat Doc...	8/3/2018 1:14	88D62D94
E02.pdf	225.656	191.817	Adobe Acrobat Doc...	22/3/2018 19:46	4799391B
E03.pdf	134.471	915.445	Adobe Acrobat Doc...	18/12/2017 14:07	A9E9388E
E04.pdf	45.911	38.172	Adobe Acrobat Doc...	13/11/2017 9:01	4799095A
E05.pdf	225.656	191.817	Adobe Acrobat Doc...	5/3/2018 10:38	4799391B
E06.pdf	225.656	191.817	Adobe Acrobat Doc...	6/3/2018 8:51	4799391B
E07.pdf	275.120	243.278	Adobe Acrobat Doc...	22/3/2018 20:08	6140E40E
E08.pdf	225.656	191.817	Adobe Acrobat Doc...	6/3/2018 8:51	4799391B
E09.pdf	225.656	191.817	Adobe Acrobat Doc...	6/3/2018 14:28	4799391B
E10.pdf	526.941	484.279	Adobe Acrobat Doc...	8/3/2018 14:29	0F118A04
E11.pdf	362.778	368.218	Adobe Acrobat Doc...	21/11/2017 15:01	80765139
E12.pdf	594.228	580.118	Adobe Acrobat Doc...	6/3/2018 9:19	14E1772B
E13.pdf	275.120	243.278	Adobe Acrobat Doc...	22/3/2018 20:08	6140E40E
E14.pdf	267.489	238.770	Adobe Acrobat Doc...	22/3/2018 19:44	FC9A6325
E15.pdf	1.351.521	1.236.664	Adobe Acrobat Doc...	22/3/2018 19:44	1546C44E
icfm.pdf	267.489	238.770	Adobe Acrobat Doc...	22/3/2018 19:44	FC9A6325
icfm.pdf	225.656	191.817	Adobe Acrobat Doc...	5/3/2018 10:38	4799391B
icfm.pdf	225.656	191.817	Adobe Acrobat Doc...	6/3/2018 8:51	4799391B
pa.pdf	406.869	526.675	Adobe Acrobat Doc...	5/3/2018 14:31	A12C484B
pac.pdf	225.656	191.817	Adobe Acrobat Doc...	22/3/2018 19:46	4799391B
pat.pdf	225.656	191.817	Adobe Acrobat Doc...	8/3/2018 14:28	4799391B
tbla.pdf	225.656	191.817	Adobe Acrobat Doc...	5/3/2018 10:39	4799391B

Figura 54. Instanciación HEIII312

En la tabla 9 establece los parámetros necesarios para ejecutar la hipótesis HEIII312, en la cual se puede visualizar la lista de materiales experimentales que se utilizan para la replicación de este experimento.

En la figura 54 se puede observar que la lista de materiales descargados por la aplicación “GESPROEXP” es igual a la lista que proporcionó el experto en replicación.

4.2.3. Caso de Prueba HEIII313

Tabla 10. Prueba de instanciación HEIII313

Experimento: Técnicas de Evaluación de Código		
Hipótesis: HEIII313		
	Criterio	Resultado
Productos Experimentales	E12	Observar Figura 55
	E13	
	E22	
	E10	
	E20	
	E01ntree	
	E02ntree	
	E03ntree	
	E04ntree	
	ICFBntree	
	ICFMntree	
	PEntree	
	E01nametbl	
	E02nametbl	
	E03nametbl	
	E04nametbl	
	ICFMnametbl	
	PEnametbl	
	E01cmdline	
	E02cmdline	
	E03cmdline	
	E04cmdline	
	ICFMcmdline	
	PEcmdline	
	Tabla de diseño	

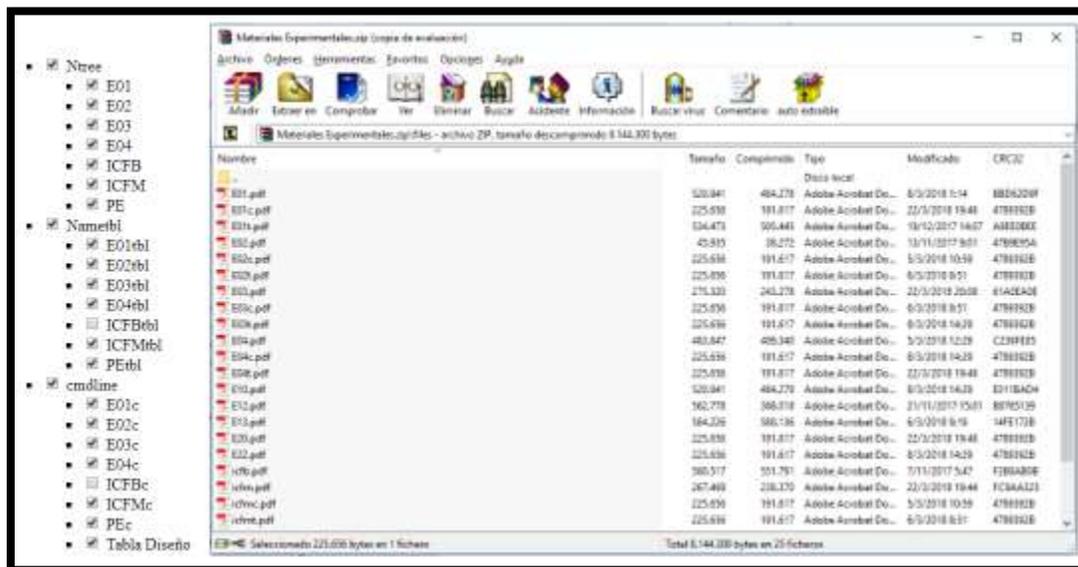


Figura 55. Instanciación HEIII313

En la tabla 10 establece los parámetros necesarios para ejecutar la hipótesis HEIII313, en la cual se puede visualizar la lista de materiales experimentales que se utilizan para la replicación de este experimento. Mientras, en la figura 55 se puede observar que la lista de materiales descargados por la aplicación “GESPPOEXP” es igual a la lista que proporcionó el experto en replicación.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- La construcción del marco teórico permitió establecer los procesos y técnicas para desarrollar una LPS.
- No existe una herramienta que permita integrar el proceso de ingeniería de dominio e ingeniería de aplicación en LPS.
- La comparación de herramientas para modelar una LPS permitió la selección de FeatureIDE, ya que no se necesita de licencia para su uso, y permite la integración con otras herramientas como: S.P.L.O.T.
- Aplicando el método FODA y la herramienta FeatureIDE, fue posible generar el modelo de dominio a partir de la matriz de Técnicas de Evaluación de código se realizó la instanciación de productos experimentales en base a la hipótesis del experimento.
- Se seleccionó la base de datos Realtime de Firebase por su estructura tipo árbol, que se adaptaba a las características del modelo de dominio.
- Al aplicar las pruebas de instanciación se pudo comprobar que los productos experimentales descargados en el módulo de configuración de experimentos de la aplicación web son iguales a los productos utilizados en la ejecución del experimento Técnicas de Evaluación de Código.
- Se desarrolló la aplicación web que permite gestionar los productos experimentales aplicando el paradigma LPS.

5.2. Recomendaciones

- Es recomendable utilizar el paradigma LPS cuando se requiere generar nuevos producto software a partir de la reutilización de elementos comunes.
- Para realiza el diseño de la LPS se recomienda el uso de fuentes primarias y tener buena comunicación con las personas que se encuentran inmiscuida en el proyecto.
- Se recomienda configurar el acceso multiorigen (CORS) para que no exista errores al momento de descargar los productos experimentales.
- Se recomienda utilizar herramientas libres como FeatureIDE para modelar LPS, ya que disponen de ejemplos, documentación y foros en los cuales guiarse y solventar cualquier duda.
- En el proceso de desarrollo es recomendable utilizar un software de control de versiones como Git, para llevar el control de los cambios realizados al código de la aplicación.

REFERENCIAS BIBLIOGRÁFICAS

- Amatriain, H. (2014). Diseño de Experimentos en Ingeniería de Software. *Universidad Nacional de Lanús*.
- ANSI/IEEE. (1983). STD 828: Standard for Software Configuration Management Plan. *IEEE*.
- Basili, V. R. (1993). The experimental paradigm in software engineering. *In Experimental Software Engineering Issues: Critical Assessment and Future Directions*, 1-12.
- Basili, V., Selby, R., & Hutchens, D. (1986). Experimentation in software engineering. *IEEE Trans. Softw. Eng.*
- Bastarrica, C. (2002). Arquitectura base en una línea de productos de software. *Universidad de Chile Departamento de Ciencias de la Computación. CEIS, Centro Experimental de Ingeniería de Software*.
- Bendix, L., Gradén, J., & Stahl, A. (2009). A configuration management perspective on composing software product lines. *Department of Computer Science, Lund University, Lund, Sweden, Tech.Rep.*
- Bosch, J. (2002). Maturity and evolution in software product lines: Approaches, artefacts and organization. *In Software Product Lines*, 257-271.
- Brooks, A., Daly, J., Miller, J., Roper, M., & Wood, M. (1996). *Replication of experimental results in software engineering*. ISERN Report.
- Clements, P., & Northrop, L. (2001). Software product lines: Practices and Patterns. *Addison-Wesley*.
- Clements, P., & Northrop, L. (2003). *A Framework for Software Product Line*. Pittsburgh, USA: Carnegie Mellon, Software Engineering.
- Clements, P., Jones, L., Northrop, L., & McGregor, J. (2005). Project management in a software product line organization. *IEEE Software*, 54-62.
- Collins, H. (1975). The seven sexes: A study in the sociology of a phenomenon, or the replication of experiments in physics. *Sociology* 9, 205-224.
- Eslava Muñoz, V. J. (2012). *HTML, presente y futuro de la web*. España.
- Espinosa, E. (2014). *Gestión de configuración y línea de productos para mejorar el proceso experimental en Ingeniería del Software*. Madrid: Universidad Politécnica de Madrid.

- Fernández, E. (2012). Proceso de agregación para estudios experimentales en Ingeniería del Software. *Universidad Nacional de la Plata*.
- Firestore. (2016). *Documentación Firestore*. Recuperado el 5 de julio de 2018, de <https://firebase.google.com/docs/?hl=es-419>
- Fonseca, C. (2012). Definition of a support infrastructure for replicating and aggregating families of software engineering experiments. *International Doctoral Symposium on empirical Software Engineering*.
- Glass, R., & Vessey, I. (2002). Research in software engineering: an analysis of the literature. *Information and Software technology* 44, 491-506.
- Gomaa, H. (2004). Designing Software Product Line with UML. *Software Engineering Workshop Tutorial*, Pref-1.
- Gómez, O., Juristo, N., & Vegas, S. (2010). Replication types in experimental disciplines. In 4th International Symposium on Empirical Software Engineering and Measurement (ESEM'2010). *Anonymous Bolzan*, 71-75.
- Gómez, O., Ucán, J., & Gómez, G. (2013). *Aplicación del proceso de experimentación a la Ingeniería de Software*. México,D.F.: Universidad Autónoma de Yucatán.
- IEEE. (1990). *IEEE Standard Glossary of Software Engineering Terminology*. IEEE std 610.12-1990.
- IEEE/EIA. (197). *IEEE 12207: Standard for Information Technology Software life cycle processes Implementation considerations*. USA: Piscataway.
- Juristo, N., & Moreno, A. (2001). Basics of Software Engineering Experimentation. *Kluwer Academic Publishers*.
- Juristo, N., & Vega, S. (2009). *Using differences among replications of software engineering experiments to gain knowledge*. International Symposium on Empirical Software Engineering and Measurement .
- Kang, K. C., Cohen, S. G., Hess, J., Novak, W., & Peterson, A. (1990). Feature-oriented domain analysis (FODA) feasibility study. *Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst*.
- Kästner, C., Thüm, T., Wielgorz, F., Saake, G., Leich, T., Feigenspan, J., & Apel, S. (2009). FeatureIDE: A tool framework for feature-oriented software development. *IEEE*, 611-614.
- Krueger, C. (2006). Introduction to the Emerging Practice Software Product Line. *METHODS & TOOLS*, 14(3).

- Linda, M., Clements, P., & Northrop, L. (2012). *A Framework for Software Product Line V. 5*. Pittsburgh, USA: Carnegie Mellon, Software Engineering.
- Mendonça, M., Branco, M., & Cowan, D. (2009). S.P.L.O.T. - Software Product Lines Online Tools. *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA*, 761-762.
- Miranda, M., Casas, S., & Marcos, C. (2015). Análisis de Desarrollo de Software Orientado a Feature - Línea de Producto de Software para Aplicaciones de TVDI. *Informes Científicos-Técnicos UNPA*, 167-195.
- Molina Caballero, J., & Baena Espejo, L. (2007). *Implantación de Aplicaciones Informáticas de Gestión*. Madrid. España: Editorial Visión Net.
- PURE::SYSTEMS. (2013). *Pure::variants Variants Management*. Recuperado el 22 de junio de 2018, de <https://www.phaedsys.com/pressreleases/pressreleasesdata/pure/purepresentation.pdf>
- Salazar, J. (2009). *Herramienta para el modelado y configuración de modelos de características*. Málaga.
- Salguero Espinosa, A. G., & Salguero Espinosa, F. P. (2015). *Diseño de una LPS (Línea de Productos de Software) para optimizar los productos intermedios y finales del proceso de desarrollo de software en la empresa CLOUDSTUDIO Servicios de Tecnología Informática Cía. Ltda. de la ciudad de Quito*. Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología. Universidad de las Fuerzas Armadas ESPE. Extensión Latacunga.
- Shadish, W., Cook, T., & Campbell, D. (2002). Experimental and quasi-experimental designs for generalized causal inference.
- Shaw, M. (1990). *Prospects for an engineering discipline of software*. IEEE Softw. 7.
- Shull, F., Carver, J., Vega, S., & Juristo, N. (2008). The role of replications in Empirical Software Engineering. *Empirical software engineering*, 211-218.
- Shull, F., Mendonça, M., Basili, V., Carver, J., Maldonado, J., Fabbri, S., . . . Oliveira, M. (2004). Knowledge-Sharing Issues in Experimental Software Engineering. *Empirical Software Engineering*.
- Solari Buela, M. (2011). Propuesta de paquete de laboratorio para experimentos en ingeniería de software. *Universidad Politécnica de Madrid*.
- Urroz Urzúa, G. (2012). *Adaptación de Software de Aplicación al Paradigma de la Ingeniería de Línea de Productos de Software*. Recuperado el 17 de julio de 2018, de <http://www.repositorio.uchile.cl/handle/2250/104366>

- Vinga-Martins, R. (1999). Requirements Traceability for Product Lines. *Workshop on Object Technology for Product-line Architectures*.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., & Wesslén, A. (1999). *Experimentation in software engineering: An introduction*. Boston: Kluwer Academic Publishers.
- Zelkowitz, M., Wallace, D., & Binkley, D. (2003). *Experimental validation of new software technology*. River Edge, NJ, USA: World Scientific Publishing Co.

ANEXOS



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SOFTWARE

CERTIFICACIÓN

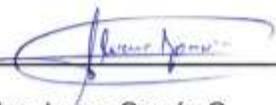
Se certifica que el presente trabajo fue realizado por el señor: **CÉSAR EDUARDO QUISHPE GUANOTÁSIG.**

En la ciudad de Latacunga, a los 31 días del mes de agosto del 2018.

Aprobado por,



PhD. Edison Espinosa G.
DIRECTOR DEL PROYECTO



Ing. Lucas Garcés G.
DIRECTOR DE CARRERA.



Abg. Darwin Albán Y.
SECRETARIO ACADÉMICO