

**TESTUIS: MARCO METODOLÓGICO DE PRUEBAS PARA PRODUCTOS
SOFTWARE DE LA UIS**

LIZETH JOHANA GARCIA GUERRERO

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD INGENIERÍA FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA**

2018

**TESTUIS: MARCO METODOLÓGICO DE PRUEBAS PARA PRODUCTOS
SOFTWARE DE LA UIS.**

LIZETH JOHANA GARCIA GUERRERO

**Trabajo de Grado para optar por el título de
Ingeniero de Sistemas e Informática**

DIRECTOR

SERGIO FERNANDO CASTILLO CASTELBLANCO

Ingeniero de Sistemas PhD

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD INGENIERÍA FÍSICO-MECÁNICAS

ESCUELA DE INGENIERÍA DE SISTEMAS

BUCARAMANGA

2018

AGRADECIMIENTOS

A Dios por darme la perseverancia para desarrollar este proyecto.

A mi director de proyecto, Sergio Fernando Castillo, por la paciencia, colaboración y confianza, que me permitieron realizar este proyecto de grado.

A mi Madre Myriam Guerrero por la compañía, los consejos y el apoyo brindado durante toda mi vida.

A mi Amigo Ariel Acuña que contribuyó en este sueño apoyándome y acompañándome durante este proceso.

TABLA DE CONTENIDO

INTRODUCCIÓN	13
1. DESCRIPCION DE PROYECTO.....	15
1.1 PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA.....	15
1.2 OBJETIVOS.....	16
1.2.1 OBJETIVO GENERAL.....	16
1.2.2 OBJETIVOS ESPECIFICOS.....	16
1.3 ALCANCE	17
2 MARCO TEORICO Y HERRAMIENTAS.....	18
2.1 MARCO TEORICO	18
2.1.1 Marco metodológico	18
2.1.2 Ciclo de vida	18
2.1.3 Verificación y Validación (V&V)	20
2.1.4 Pruebas de software.....	23
2.1.5 Estándares de Verificación y Validación	24
2.2 ESTADO DEL ARTE.....	25
2.3 HERRAMIENTAS.....	27
2.3.1 PHP	28
2.3.2 HTML.....	28
2.3.3 CSS	29
2.3.4 XAMPP	29
2.3.5 VISUAL STUDIO CODE	29
2.3.6 LARAVEL.....	30
2.3.7 BOOTSTRAP.....	30
3 MARCO METODOLOGICO	31
3.1 DESCRIPCION DEL MARCO METODOLOGICO	31
3.1.1 Ciclo de Vida.....	31
3.1.2 Modelo en W propuesto.....	32
3.1.3 Equipos de trabajo del marco metodológico	33

3.1.4	Descripción de las principales etapas.....	34
4	DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS	38
4.1	REQUERIMIENTOS DEL SISTEMA.....	38
4.2	CASOS DE USO ASOCIADOS.....	39
5	DISEÑO DE SOFTV&V.....	43
5.1	DIAGRAMAS MER.....	43
5.1.1	Tablas del Sistema	43
5.1.2	Tablas de Análisis y Revisión de Análisis	44
5.1.3	Tablas de Diseño y Revisión Diseño	46
5.1.4	Tablas de Diseño Detallado y Revisión de Diseño Detallado	48
5.1.5	Tablas de Pruebas y Depuración y Cambios.....	49
5.2	ARQUITECTURA.....	50
5.3	INTERFACES PRINCIPALES DE SOFTV&V	51
5.3.1	Inicio De Sesión.....	51
5.3.2	Registro de Nuevo Usuario.....	51
5.3.3	Gestión de Proyectos.....	52
5.3.4	Modelo W.....	52
5.3.5	Análisis	53
5.3.6	Revisión Análisis.....	53
5.3.7	Diseño.....	54
5.3.8	Revisión de Diseño	55
5.3.9	Diseño Detallado	55
5.3.10	Revisión de Diseño Detallado.....	56
5.3.11	Pruebas	57
5.3.12	Depuración y Cambios	57
6	PRUEBAS.....	59
6.1	PRUEBAS FUNCIONALES	59
7	CONCLUSIONES	65
8	RECOMENDACIONES	66

BIBLIOGRAFÍA.....67

LISTADO DE FIGURAS

Figura 1 Ciclo de vida en cascada	19
Figura 2 Modelo en V.....	22
Figura 3 Modelo de W de Paul Herzlich.....	23
Figura 4 modelo en W de MADEJA	26
Figura 5 Etapas del ciclo de vida Vs Pruebas.....	32
Figura 6 Modelo W.....	32
Figura 7 Registro y Autenticación	39
Figura 8 Caso de uso: Gestión de Proyectos	40
Figura 9 Registro de usuarios	41
Figura 10 Tablas del Sistema	43
Figura 11 Tablas etapa de Análisis.....	44
Figura 12 Tablas etapa Revisión de Análisis	46
Figura 13 Tablas etapa de Diseño	47
Figura 14 Tablas etapa Revisión Diseño	48
Figura 15 Tablas etapas Diseño Detallado y Revisión de Diseño Detallado	49
Figura 16 Pantalla de Inicio de Sesión.....	51
Figura 17 Pantalla de Registro de Usuario	51
Figura 18 Pantalla de Gestión de Proyectos.....	52
Figura 19 Pantalla de Modelo W.....	52
Figura 20 Pantalla Etapa Análisis	53
Figura 21 Pantalla Etapa Revisión de Análisis	54
Figura 22 Pantalla Etapa Diseño	54
Figura 23 Pantalla Etapa Revisión Diseño.....	55
Figura 24 Pantalla Etapa Diseño Detallado	56
Figura 25 Pantalla Etapa Revisión de Diseño Detallado.....	56
Figura 26 Pantalla Etapa Pruebas	57
Figura 27 Pantalla Etapa Depuración y Cambios	58

LISTADO DE TABLAS

Tabla 1 Estándares de validación y verificación	24
Tabla 2 Caso de uso: iniciar sesión	39
Tabla 3 Caso de uso: Registrarse.....	39
Tabla 4 Caso de uso: iniciar sesión	40
Tabla 5 Caso de uso: Crear proyecto	40
Tabla 6 Caso de uso: Gestionar proyecto.....	41
Tabla 7 Caso de uso: Eliminar proyecto	41
Tabla 8 Caso de uso: Agregar Usuario.....	42
Tabla 9 Caso de uso: Asignar Rol	42
Tabla 10 Caso de uso: Eliminar proyecto	42
Tabla 11 Pruebas Pantalla de Ingreso.....	59
Tabla 12 Pruebas Pantalla Gestión Proyectos	59
Tabla 13 Pruebas Pantalla Modelo W.....	60
Tabla 14 Pruebas Etapa Análisis.....	60
Tabla 15 Pruebas Etapa Revisión de Requisitos	61
Tabla 16 Pruebas Etapa Diseño	61
Tabla 17 Pruebas Etapa Revisión Diseño	62
Tabla 18 Pruebas Etapa Diseño Detallado	62
Tabla 19 Pruebas Etapa Revisión Diseño Detallado	63
Tabla 20 Pruebas Etapa de Pruebas	63
Tabla 21 Pruebas Etapa Depuración y Cambios	64

RESUMEN

TITULO: TESTUIS: MARCO METODOLÓGICO DE PRUEBAS PARA PRODUCTOS SOFTWARE DE LA UIS

AUTOR: LIZETH JOHANA GARCIA GUERRERO**

PALABRAS CLAVE: modelo w, testing, pruebas, verificación, validación, testing temprano, calidad, ciclo de vida.

DESCRIPCIÓN: Las tecnologías de la información están cada vez más involucradas en todos los aspectos de la sociedad, desde sistemas para manejo de la salud hasta software para el entretenimiento, convirtiéndose en algo fundamental de la vida diaria de las personas. Se hace necesario que el desarrollo de software emplee procesos que garanticen un producto de calidad que esté validado y verificado, es decir, que el producto se construye correctamente y que se construye el producto correcto. El presente proyecto de grado contribuye a las actividades de verificación y validación de productos software a través de la elaboración de un marco metodológico, el cual está basado en el modelo W propuesto por Paul Herzlich en 1993, y se le adicionan actividades de testing temprano después de cada etapa del ciclo de vida del software. Se definen y exponen un conjunto de prácticas y plantillas que ayudan a desarrollar y gestionar correctamente las tareas relacionadas con la producción de software, en miras de garantizar la calidad del producto software.

Adicionalmente, se desarrolló un prototipo de aplicación web, con el que se gestiona los documentos de especificación y planeación de pruebas resultantes de cada etapa de los ciclos de vida del software, ejemplificando cómo se usaría del marco metodológico en desarrollo de software iterativo y en cascada.

* Trabajo de Grado

** Facultad Ingeniería Físico-Mecánicas. Escuela Ingeniería de Sistemas. Director Ing. Sergio Fernando Castillo Castelblanco.

ABSTRACT

TITLE: TESTUIS: METHODOLOGICAL FRAMEWORK OF TESTS FOR SOFTWARE PRODUCTS OF THE UIS*

AUTHOR: LIZETH JOHANA GARCIA GUERRERO**

KEY WORDS: w model, testing, software tests, verification, validation, early testing, quality, life cycle.

DESCRIPTION:

Information Technologies are increasingly more involved with all aspects of society, from systems used for health insurance to software for entertainment purposes, they have become part in people's daily lives. That is why it is necessary for software development to use processes that guarantee a quality product that is validated and verified, i.e., to have a product built right and to build the right product. The following bachelor's degree thesis contributes to the activities of validation and verification of software products through the elaboration of a methodological framework, which is based on the W model proposed by Paul Herzlich in 1993, and with early testing activities added after each stage of the software life cycle. It is defined and shown a set of practices and templates that help to develop and manage correctly all tasks related with software production, in order to guarantee the quality of the developed software product.

Additionally, a prototype web application was developed. It manages the specification documents and planning of tests which are the result of finalizing each stage of the software life cycles. It was developed with the purpose of showcasing an example of how the methodological framework would be used in iterative and cascading software development.

* Bachelor Thesis

** Facultad Ingeniería Físico-Mecánicas. Escuela Ingeniería de Sistemas. Director Ing. Sergio Fernando Castillo Castelblanco.

INTRODUCCIÓN

Diversos estudios, como el presentado en el informe de desempeño del sector software¹ muestran que la demanda software se ha incrementado de manera positiva, de modo que las tecnologías de la información están cada vez más involucradas en todos los aspectos de la sociedad. Ante esta demanda, las empresas desarrolladoras de software deben emplear procesos que garanticen un producto confiable para sus clientes.

En la industria los productos software pueden verse inmersos en problemas que pueden ser atribuidos al permanente cambio de condiciones y a su obsolescencia entre otros, que los llevan a fracasar o a simplemente no terminar con el éxito que se esperaría. Esto hace necesario incrementar la calidad de los procesos implementados, obteniéndose por ende una mayor calidad en los productos software. La mejora del proceso de pruebas software aporta un buen número de beneficios que influyen en la calidad de los productos y por tanto en el éxito de los proyectos.

Uno de los más importantes procesos es el de verificación y validación con el cual se asegura que el software que se está desarrollando está acorde con sus especificaciones y suple las necesidades del cliente. Es relevante llevar a cabo este proceso para mejorar la calidad de los productos durante todo el ciclo de vida software, se debe intentar corregir cualquier posible fallo para alcanzar de cierto modo un grado de perfección y así asegurar que el sistema será útil para el entorno en el cual es requerido.

¹ Desempeño del sector software 2012-2014 (Informe) [En línea]. Superintendencia de sociedades (Recuperado en 09 de agosto 2018). Disponible en: https://www.supersociedades.gov.co/delegatura_aec/estudios_financieros/Documents/Sectores%20Economicos/Estudio%20Sectorial%20Software%202015.pdf

Para alcanzar tal éxito se hace necesario disponer de un marco metodológico que generara opciones para todo proceso de desarrollo considerando algunos criterios, los cuales deben estar bien definidos, formalizados; que va a permitir ejecutar un proceso de pruebas eficiente. El modelo ha de ser escalable de modo que cualquier organización pueda implementarla acorde a sus necesidades.

El presente proyecto de grado contribuye a las actividades de verificación y validación de productos software a través de la elaboración de un marco metodológico y un prototipo de aplicación web en el que se definan y expongan un conjunto de prácticas y plantillas que ayuden a desarrollar y gestionar correctamente las tareas relacionadas con el proceso de pruebas de software y de esta manera incrementar la calidad de los productos.

1. DESCRIPCIÓN DE PROYECTO

1.1 PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

En la actualidad para obtener un buen control de calidad de un producto software hay que enfocarse en los procesos de Validación y Verificación ya que tienen como objetivo satisfacer los requisitos del usuario para obtener la calidad esperada, pero muchas veces no son considerados como lo afirman Tokar y Mankefort en su estudio: “Estadísticamente el 60% de los desarrolladores afirma que las actividades de Verificación y Validación no son ejecutadas o son las primeras que abandonan cuando se encuentran bajo presión”²

Aunque existe mucha información sobre la mejora de calidad software son difíciles de interpretar ya que dicen que se debe hacer, pero no especifican como implementarlo, no cuentan con plantillas, documentos de apoyo y no identifican cuales serían los roles básicos o se necesita una fuerte inversión de dinero y tiempo para realizar las actividades de Verificación y Validación por eso es que esta actividad la termina realizando el mismo desarrollador y al final del proyecto, así es que se llega a un ambiente caótico e incontrolado y se obtiene un producto software con baja calidad, aumento de re trabajo, insatisfacción por parte del cliente y altos costos.

Por eso este proyecto de grado mejora las actividades de verificación y validación de productos software través de la elaboración de un marco metodológico que por medio de un prototipo en un sitio web se podrá implementar de una manera más

² A survey on testing and reuse, TOKAR R., MANKEFORTS S., 2003 [En línea]. IEEE International Conference on Software (Recuperado en 09 de agosto 2018). Disponible en: <https://ieeexplore.ieee.org/document/1245437>

comprensible y ágil el conjunto de prácticas, normas y plantillas para así incrementar la calidad de los productos.

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL. Plantear una propuesta de un marco metodológico para el apoyo de pruebas que permita la Verificación y Validación de un producto software.

1.2.2 OBJETIVOS ESPECIFICOS

- Definir un marco metodológico utilizando un conjunto de normas y plantillas que ayuden a desarrollar y gestionar las tareas relacionadas con las actividades de Verificación y Validación de productos software.

- Diseñar los módulos del marco metodológico que apoyan los procesos de Verificación y Validación de un producto software, y su respectivo modelo de información. Las principales funcionalidades a incluir son:
 - Verificación y Validación en la etapa de Análisis
 - Verificación y Validación en la etapa de Diseño
 - Verificación y Validación en la etapa de Pruebas

- Implementar un prototipo de aplicación web para gestionar adecuadamente la información y los recursos disponibles para el marco metodológico propuesto.

- Realizar pruebas del marco metodológico y del prototipo desarrollado.

1.3 ALCANCE

El alcance de este trabajo de investigación es desarrollar una propuesta de un Marco Metodológico para el apoyo de pruebas que por medio del proceso de Verificación y Validación en los proyectos de desarrollo software y basándose en estándares de calidad más representativos del mercado aumentara la calidad.

Asimismo, este proyecto propone un prototipo de aplicación web que busca dar a conocer de una manera más comprensible a los usuarios el marco metodológico.

2 MARCO TEORICO Y HERRAMIENTAS

En este capítulo se muestra una definición más detallada sobre los principales conceptos que se manejarán en este proyecto, como son los conceptos de: verificación, validación, pruebas de software, marco metodológico y las herramientas aplicadas en la ejecución de este proyecto.

2.1 MARCO TEORICO

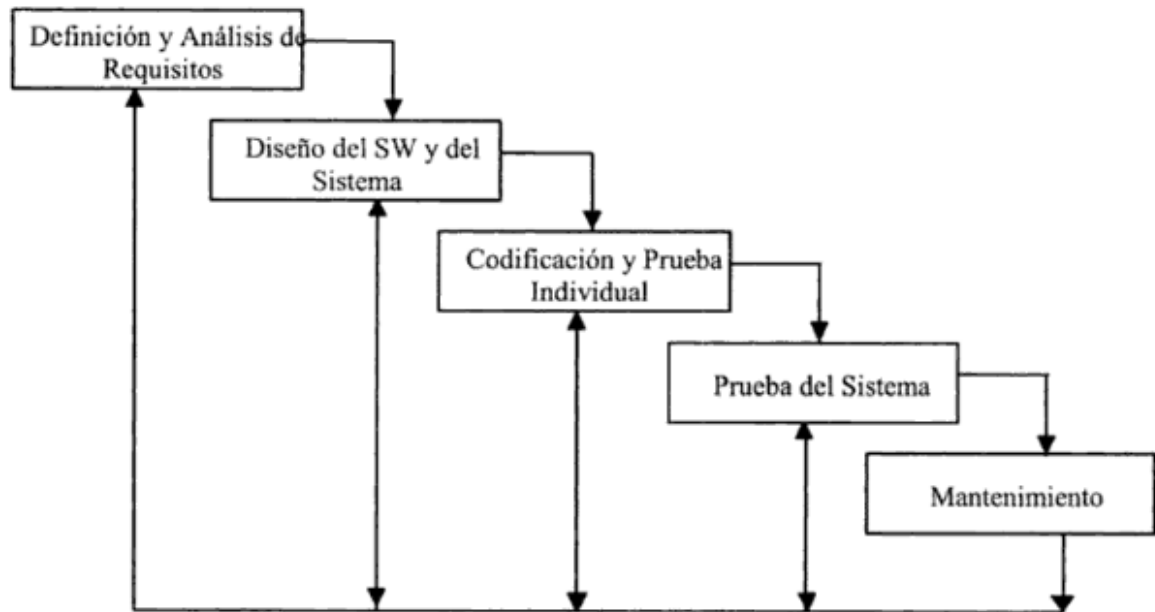
2.1.1 Marco metodológico. Es una investigación que debe comprenderse como una herramienta fundamental para indagar en el contexto de estudio, que consta de una serie de pasos en general pero que varía según el contexto en el que se está trabajando, plantear una secuencia de pasos correctos nos lleva a extraer los resultados esperados dentro de la investigación.

2.1.2 Ciclo de vida. Es el conjunto de fases por las que pasa el sistema que se está desarrollando desde que se tiene una idea inicial hasta que el software es eliminado o sustituido.

Un ciclo de vida para un proyecto se compone de fases consecutivas en las cuales se planifican las tareas a desarrollar.

El primer modelo de procesos de desarrollo de software que se publicó se derivó de procesos más generales propuestos por Royce en 1970 y desde entonces el ciclo de vida en cascada es uno de los más desarrollados y utilizados en las últimas décadas.

Figura 1 Ciclo de vida en cascada



Fuente: Ciclo de vida en cascada [en línea] (recuperado el 20 de agosto de 2018).

Disponible en:

<https://books.google.com.co/books?id=MXTI43ThoS4C&printsec=frontcover#v=onepage&q&f=false>

En la Figura 1 se muestra la organización más habitual de las fases de desarrollo, aunque en algunos textos se puedan encontrar variaciones en las fases eso no afecta al desarrollo del software si no a la manera de agrupar dichas fases.

Esta propuesta de marco metodológico se basa en desarrollar pruebas tempranas en algunas fases del ciclo de vida y no solo en la fase de pruebas del sistema.

2.1.3 Verificación y Validación (V&V). La verificación y validación son un conjunto de procesos de comprobación y análisis para asegurar que se cumplan las especificaciones dadas y las necesidades de los clientes, tal como lo define Sommerville “La verificación y la validación (V&V) es el nombre dado a estos procesos de análisis y pruebas. La verificación y validación tienen lugar en cada etapa del proceso de software. V&V comienza con revisiones de los requerimientos y continúa con revisiones del diseño e inspecciones de código hasta la prueba del producto”³

Una manera más informal de comprender la definición de V&V la propone Barry W. Boehm

- “Verificación: ¿Estoy construyendo el producto correcto?”
- Validación: ¿Estoy construyendo el producto correctamente?⁴

Es decir que el proceso de verificación permite encontrar errores y comprobar que el software satisface los requerimientos definidos; mientras que el proceso de validación busca asegurar que el software satisfaga las expectativas del usuario ya que las especificaciones del software no siempre reflejan las necesidades reales de usuario.

2.1.3.1 Beneficios del proceso de la Verificación y Validación

- “Asegurar los requerimientos del usuario.
- Remover los defectos del producto fuera del ciclo de vida del proyecto, reduciendo el volver a trabajar sobre el mismo aspecto, y reduciendo los costos por la baja calidad.

³SOMMERVILLE, Ingeniería del software, 2005. p. 472.

⁴BARRY W. Boehm, Guidelines for Verifying and Validating Software Requirements and Design Specifications, 1984. p. 75.

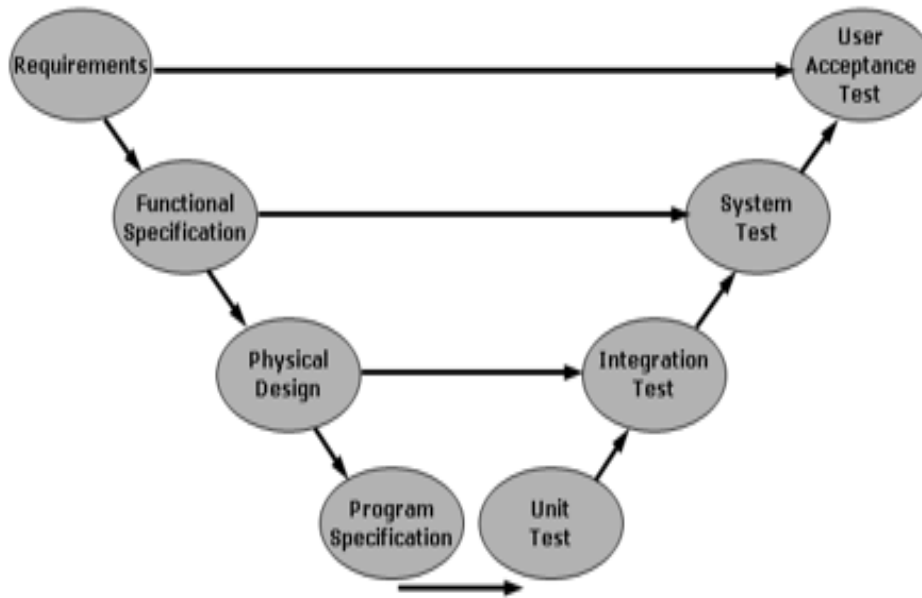
- Asegura que las necesidades de los usuarios son comprendidas y asegura que los productos satisfagan el entorno previsto para el que fueron desarrollados.
- Mejorar la calidad de los procesos y los productos.
- Mejorar la productividad y permite conocer de manera temprana el rendimiento”.⁵
- Mejorar la comprensión de la gestión de riesgos de procesos y productos.
- Proporcionar evidencia objetiva de conformidad del producto para apoyar al proceso de certificación.
- Soporte a las actividades de mejora de procesos.

2.1.3.2 Modelo en V y W de la Verificación y Validación. El modelo en V como se observa en la Figura 2 es muy sencillo y fácil de comprender ya que promueve la noción temprana del testing temprano es decir que intenta encontrar fallos tempranos de una manera secuencial y por niveles

La desventaja del modelo es que trata las pruebas como una “puerta trasera” que sería la parte derecha del modelo es decir las actividades de prueba comienzan por primera vez después de la implementación por eso puede crearse la idea de que después de la fase de implementación el producto está listo y puede ser entregado.

⁵ PUELLO, O., Modelo de Verificación y Validación Basado en CMMI. *Investigación e Innovación en Ingenierías*, 2013 p. 1.

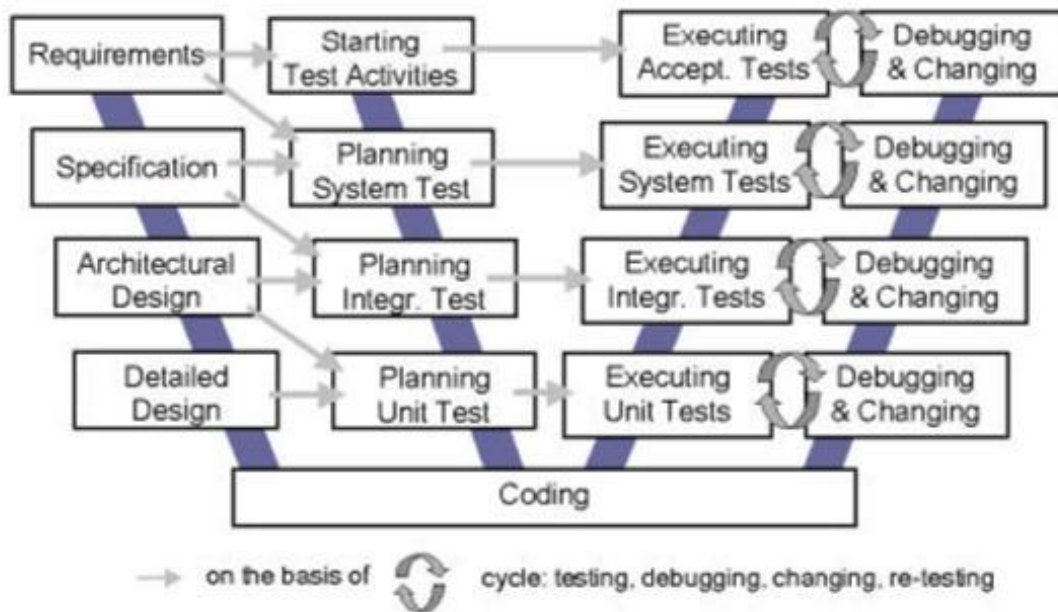
Figura 2 Modelo en V



Fuente: Modelo en V [en línea] (recuperado el 20 de agosto de 2018). Disponible en: <http://www.gerrardconsulting.com/?q=node/531>

Una mejora del modelo en V es el modelo en W que fue introducido por Paul Herzlich en 1993 este modelo implementa en cada etapa del ciclo de vida una actividad de prueba, permite ajustarse fácilmente si la organización tiene diferentes etapas de desarrollo y las tareas de depuración y cambio son claras por eso las pruebas son más eficaces y lo podemos observar en la Figura 3. Por eso para el marco metodológico propuesto se implementó el modelo en W.

Figura 3 Modelo de W de Paul Herzlich



Fuente: SPILLNER, A., & BREMENN, H., The W-MODEL. Strengthening the bond between development and test. In *Int. Conf. on Software Testing, Analysis and Review* p. 15-17

2.1.4 Pruebas de software. Son actividades claves para que los procesos de verificación y validación sean realizados con éxito, ya que ayudan a entregar un producto con la calidad suficiente para lograr una satisfacción en los clientes y alcanzar la certeza de que el producto cumple con los requisitos establecidos. En este sentido las pruebas proporcionan un nivel de confianza en el software.

El principal objetivo de las pruebas es el de encontrar errores en el programa, tantos como sea posible.

Entorno a este tema surgen varios conceptos que a menudo generan confusión. Por ellos, el estándar IEEE/ANSI 610.12-1990 establece:

- Falta: acción humana que produce un resultado incorrecto.
- Defecto: paso, proceso o definición de datos incorrectos en una aplicación a consecuencia de una falta.
- Fallo: resultado incorrecto, es el resultado de un defecto.

- Error: el valor de un resultado incorrecto.

2.1.4.1 Testing Temprano. El objetivo Principal del testing temprano es reducir costos a través de la detección de errores en las fases tempranas del ciclo de vida del producto. Para ejecutarlo es necesario tener la documentación necesaria y definir el plan de pruebas de aceptación, con ello podemos tener una trazabilidad de requerimientos para prevenir los defectos que aparezcan en fases posteriores del producto.

2.1.5 Estándares de Verificación y Validación. En la siguiente tabla se muestra los grupos y estándares más relevantes para

Tabla 1 Estándares de validación y verificación

Estándares	Descripción
IEEE 830-1984	Este estándar presenta una guía para escribir requerimientos específicos software. Por medio de la descripción y el contenido adquiere las cualidades necesarias para una buena especificación de requisitos de software.
IEEE 730-2002	Este estándar provee los requerimientos formales mínimos para la realización de planes de aseguramiento de calidad de software para ser ejecutados en el momento en que el software es desarrollado o mantenido.
IEEE 1028-1997	Este estándar provee modelos para los cinco tipos distintos de revisión: revisiones gerenciales, revisiones técnicas, inspecciones, y auditorias.

2.2 ESTADO DEL ARTE

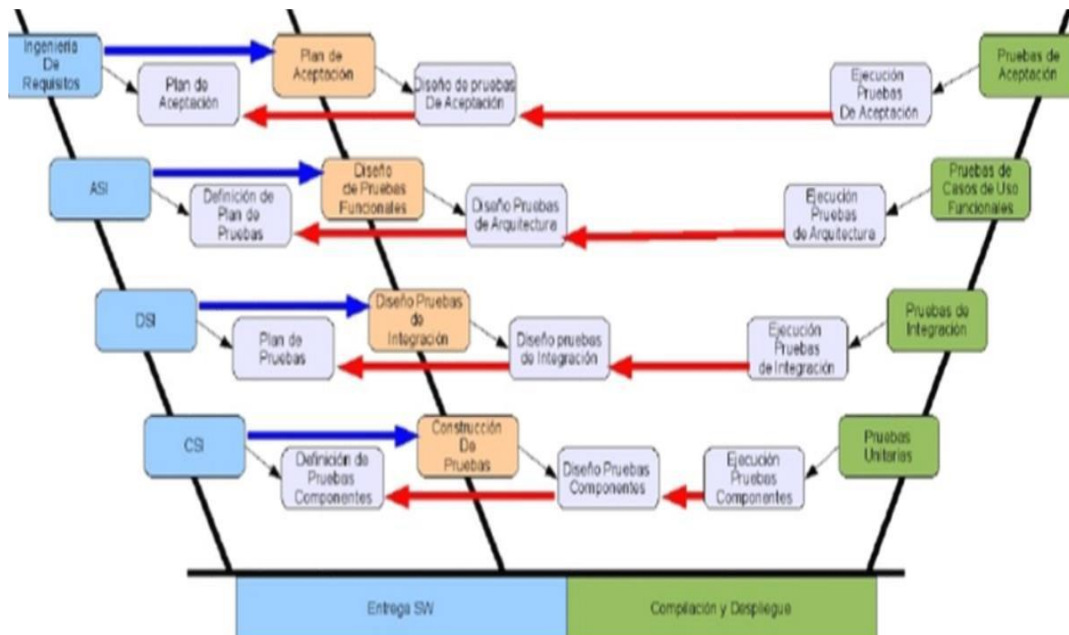
En la actualidad hay diversos modelos de referencia que nos dan a conocer la implementación de un marco metodológico, esto permite una mejora de procesos dentro de las organizaciones. Es decir, los modelos facilitan al marco metodológico un conjunto de fortalezas y debilidades de los procesos y las acciones a realizar para la mejora de estos.

- **Marco de Desarrollo de la Junta de Andalucía (MADEJA)**

La Junta de Andalucía maneja los procesos de pruebas, verificaciones y validaciones de aplicaciones y sistemas de información. MADEJA utiliza dos componentes durante los procesos de pruebas que son las directrices marcadas por esta y las especificaciones propias de cada proyecto. Durante la ejecución de los proyectos las pruebas de calidad siempre están presentes y establece un conjunto objetivo de criterios de validación y verificación de las entregas a lo largo del ciclo de vida y para ello adoptan un modelo en W⁶

⁶ Modelo en W de MADEJA [en línea] Marco de Desarrollo de Software de la Junta de Andalucía. (recuperado el 20 de agosto de 2018). Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/verificacion>

Figura 4 modelo en W de MADEJA



Fuente: Modelo en W de MADEJA [en línea] Marco de Desarrollo de Software de la Junta de Andalucía. (recuperado el 20 de agosto de 2018). Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/verificacion>

• **Marco Metodológico para la mejora de las actividades de verificación y validación de productos software**

Esta tesis doctoral realizada por Ana Sanz Esteban en el 2012 en Leganés - España elaboró un marco metodológico que defina el conjunto de prácticas eficientes para desarrollar las tareas relacionadas con la verificación y validación de productos software y formar y gestionar equipos de pruebas eficientes, es decir que tengan conocimiento acerca del trabajo que han de desarrollar y dispongan de las competencias requeridas para ello.⁷

Esta tesis centró su investigación del marco metodológico considerando dos variables *personas* y *procesos*, las cuales en la definición de un modelo de *procesos*

⁷ ESTEBAN. Ana.,2012 [en línea] Marco Metodológico para la mejora de las actividades de verificación y validación de productos software. (recuperado el 20 de agosto de 2018). Disponible en: https://e-archivo.uc3m.es/bitstream/handle/10016/15464/tesis_ana_sanz_esteban_2012.pdf

se estructura y se especifica las actividades a realizar en relación con la V&V de productos software y un modelo de competencias que defina el conocimiento y las habilidades que han de poseer las *personas* encargadas de la verificación y validación de los productos y hasta la actualidad no hay ninguna herramienta que lo implemente.

- **Marco Metodológico del proceso de verificación y validación de software para pequeñas y medianas empresas**

Esta tesis de maestría realizada por Lizette Grados Aguirre en el 2015 en Lima – Perú elaboró un Marco Metodológico del Proceso de Verificación y Validación ajustado a la realidad socio económico de las Pymes y enfocado en los procesos claves de V&V, por ello, se tomaron como referencia los modelos y estándares de calidad: IEEE Std. 1012, IEEE Std. 829, IEEE Std. 1008, en el trabajo se muestra una guía metodológica, los roles y competencias del equipo de trabajo, plantillas/listas de verificación base y las principales métricas.⁸

Este marco metodológico se basó en los modelos y estándares definidos por la IEEE, realizó comparaciones entre los modelos de calidad para determinar cuál aplicar en el ámbito socio económico de las Pymes y definir los roles y el equipo de trabajo para la ejecución de los procesos de V & V.

2.3 HERRAMIENTAS. En este capítulo se dan a conocer las herramientas aplicadas en este proyecto, con las cuales se desarrolló el prototipo web que permite acceder a la información del marco metodológico propuesto.

⁸AGURRE. Lizette.,2005 [en línea] Marco Metodológico del proceso de verificación y validación de software para pequeñas y medianas empresas. (recuperado el 20 de agosto de 2018). Disponible en: <https://core.ac.uk/download/pdf/54217032.pdf>

2.3.1 PHP. (acrónimo recursivo de PHP: Hypertext Pre-processor) es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante⁹

2.3.2 HTML. (HyperText Markup Language) es un lenguaje de marcado que los ordenadores son capaces de interpretar y que por medio de instrucciones de los desarrolladores los navegadores ejecutan para crear las páginas web.

Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas (<,>) que permiten interconectar diversos conceptos y formatos.

Por otra parte, cabe destacar que el HTML permite ciertos códigos que se conocen como scripts, los cuales brindan instrucciones específicas a los navegadores que se encargan de procesar el lenguaje. Entre los scripts que pueden agregarse, los más conocidos y utilizados son JavaScript y PHP.¹⁰

HTML es un estándar a cargo de World Wide Web Consortium (W3C)

⁹ PHP [En línea]. Wikipedia, La enciclopedia libre, 2018(Recuperado en 09 de agosto 2018). Disponible en: <https://es.wikipedia.org/wiki/PHP>

¹⁰ HTML [En línea]. Porto. J, Gardey A, 2012 (Recuperado en 09 de agosto 2018). Disponible en: <http://definicion.de/html/#ixzz40RI1BM1P>.

2.3.3 CSS. Las hojas de estilo en cascada (CSS) son elementos agregados al lenguaje HTML que tomaran en cuenta la presentación del documento o de la aplicación Web.

El Concepto ligado a CSS se encuentra en el principio de separación entre contenido y presentación a la hora de programar aplicaciones HTML. De este modo, el mismo contenido puede visualizarse según la hoja de estilo adoptada de forma diferente en distintos dispositivos.¹¹

2.3.4 XAMPP. Es un paquete de instalación independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.¹²

La ventaja de XAMPP es sólo requiere una pequeña fracción del tiempo necesario para descargar y ejecutar un archivo ZIP, tar, exe o fkl (pudiendo encontrarse o en versión completa como en una versión más ligera, que es portable).

2.3.5 VISUAL STUDIO CODE. Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias.¹³

Su principal ventaja es que al ser un editor de código fuente es compatible con varios lenguajes de programación

¹¹ CSS [En línea]. Wikipedia, La enciclopedia libre, 2018 (Recuperado en 09 de agosto 2018). Disponible en https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada

¹²XAMPP [En línea]. Wikipedia, La enciclopedia libre, 2018 (Recuperado en 09 de agosto 2018). Disponible en <https://es.wikipedia.org/wiki/XAMPP>

¹³ VISUAL STUDIO CODE [En línea]. Wikipedia, La enciclopedia libre, 2018 (Recuperado en 09 de agosto 2018). Disponible en: https://es.wikipedia.org/wiki/Visual_Studio_Code

2.3.6 LARAVEL. Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.¹⁴

El objetivo principal de laravel es ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades.

2.3.7 BOOTSTRAP. Es una biblioteca multiplataforma 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 adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.¹⁵

Su principal característica es que es compatible con la mayoría de los navegadores web

¹⁴ LARAVEL [En línea]. Wikipedia, La enciclopedia libre, 2018 (Recuperado en 09 de agosto 2018). Disponible en: https://es.wikipedia.org/wiki/Laravel#cite_note-introduccion-2

¹⁵ BOOTSTRAP [En línea]. Wikipedia, La enciclopedia libre, 2018 (Recuperado en 09 de agosto 2018). Disponible en: [https://es.wikipedia.org/wiki/Bootstrap_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework))

3 MARCO METODOLOGICO

Este Capítulo propone un Marco Metodológico de pruebas basándose en el proceso de Verificación y Validación Software en el cual se señala los estándares de calidad que sirvieron como base para la propuesta, y la descripción de las etapas propuestas con sus respectivas funcionalidades.

3.1 DESCRIPCION DEL MARCO METODOLOGICO

Lo primero que debemos considerar es que el proceso de Validación y Verificación se lleva a lo largo de todo el ciclo de vida del desarrollo y se aplica por medio de un modelo en W adaptado y modificado, pero partiendo de la base del modelo propuesto por Paul Herzlich en 1993 y que se menciona anteriormente en el marco teórico.

3.1.1 Ciclo de Vida. Ya que el ciclo de vida es el conjunto de fases por las que pasa el sistema que se está desarrollando desde que se tiene una idea inicial hasta que el software es eliminado o sustituido lo que se busca este marco metodológico es aumentar la detección de errores por medio de pruebas que se realizaran en cada etapa del ciclo de vida por medio del testing temprano y de las pruebas tradicionales como se muestra en la Figura 5

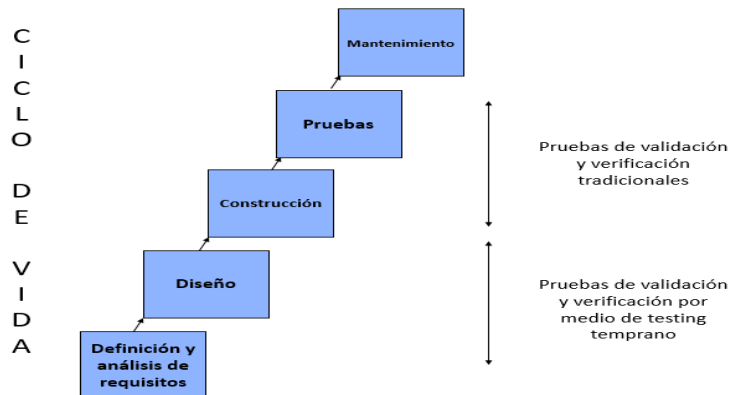
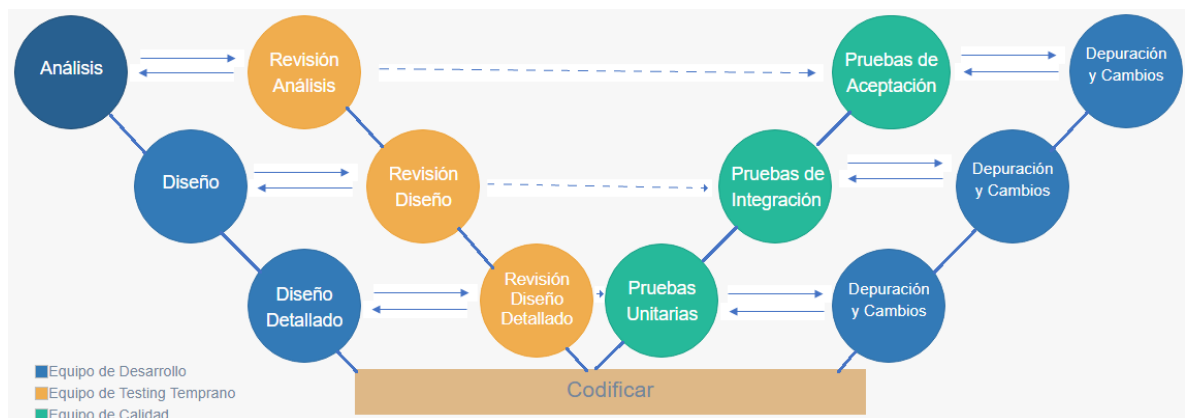


Figura 5 Etapas del ciclo de vida Vs Pruebas

En la Figura 5 de manera diagonal podemos ver las 5 etapas del ciclo de vida más comunes que son: Definición y análisis de requisitos, Diseño, Construcción, Pruebas y mantenimiento, y al lado derecho de manera vertical las pruebas de Verificación y Validación (VyV) por medio del testing temprano que inician en la etapa de Definición y análisis de requisitos y termina en la Diseño, dejando a las pruebas de VyV tradicionales ejecutarse en las demás etapas del ciclo de vida.

3.1.2 Modelo en W propuesto. Partiendo de la base del modelo propuesto por Paul Herzlich en 1993 y adaptándolo para un mayor desempeño se propone el siguiente modelo en W en donde en la Figura 6 se observa de manera general los roles de cada equipo y las etapas que posteriormente de detallan a profundidad.

Figura 6 Modelo W



El modelo W se trabaja de manera vertical, y a su vez de manera horizontal se realizan pruebas de Validación y Verificación para aumentar su calidad como se observa en la Figura 6.

De manera vertical y partiendo de izquierda a derecha están las etapas principales que son el Análisis, Diseño y Diseño Detallado representadas de color azul y al lado derecho esta su respectiva etapa de testing temprano representadas por el color

anaranjado al finalizar con estas etapas obtendremos las herramientas necesarias para codificar de manera ágil y eficiente el software

Posteriormente y una vez se tenga codificado el software pasara a la etapa de pruebas tradicionales representadas de color verde, iniciando por las Pruebas Unitarias pasando por las Pruebas de Integración y terminando en las Pruebas de Aceptación que en caso de no ser aprobadas en su primer intento deberán pasar a la etapa de Depuración y Cambios representadas de color azul hasta tener un resultado exitoso.

3.1.3 Equipos de trabajo del marco metodológico. En el marco metodológico propuesto se parte de 3 equipos de trabajo y un líder del proyecto para llevar a cabo su desempeño correctamente.

- Líder del proyecto
- Equipo de Desarrollo
- Equipo de Testing Temprano
- Equipo de Calidad

3.1.3.1 Líder del proyecto. Es el encargado de gestionar, supervisar y controlar el proyecto por medio de la planificación de este e identificando y analizando las posibles desviaciones de la planificación definida, adicionalmente debe solucionar y/o escalar cualquier problema existente en el proyecto, teniendo en cuenta que él tiene acceso a todas las etapas del proyecto.

Para este rol se recomienda estar certificado en ISTQB (International Software Testing Qualifications Board) nivel avanzado.

3.1.3.2 Equipo de Desarrollo. El equipo debe contar con personas de un perfil técnico funcional, responsable de generar un análisis en detalle de manera que cumpla las expectativas de calidad del cliente, generando los requisitos y diseños funcionales para el proyecto y así proporcionar la información adecuada para alcanzar los objetivos.

3.1.3.3 Equipo de Testing Temprano. El equipo debe contar con personas de un perfil técnico responsable de la elaboración y supervisión de las revisiones por medio de la documentación verificada, aprobación de requisitos, diseños funcionales y desarrollo de plan de pruebas para futuras pruebas tradicionales con lo cual se busca tomar acciones preventivas y aumentar la calidad del proyecto.

Se recomienda estar certificado en ISTQB (International Software Testing Qualifications Board).

3.1.3.4 Equipo de Calidad. El equipo debe contar con personas de un perfil técnico responsable de ejecutar las pruebas como fueron definidas en el plan de pruebas, documentar y reportar los resultados y hacer un seguimiento a las incidencias encontradas.

Se recomienda estar certificado en ISTQB (International Software Testing Qualifications Board).

3.1.4 Descripción de las principales etapas. A continuación, se describen las actividades de V&V en cada Etapa de SoftV&V.

3.1.4.1 Etapa de Análisis. En esta etapa las actividades se enfocan en analizar y especificar los requisitos del sistema, los cuales deben ser claros y sin ambigüedades con la finalidad de asegurar su entendimiento por parte de los demás integrantes del equipo. En esta etapa se parte de que el levantamiento de requisitos se hace basado en el STD IEEE 830-1998.

El equipo de desarrollo es el encargado de esta actividad. Al culminar el análisis, se delega al equipo de testing temprano la **Revisión del análisis**. En la revisión se verifica que los requisitos sean claros, cumplan con las necesidades del usuario y las expectativas del cliente. También se busca establecer las prioridades de cada requisito. Esta etapa se realizará tantas veces sea necesario hasta que el análisis de requisitos sea aprobado a satisfacción por el cliente.

Una vez se hayan corregido los defectos detectados y se tengan analizados los requisitos finales se realizará el plan de pruebas de aceptación en la **Revisión de análisis**, el cual es realizado por el equipo de testing temprano y debe ser validado por el Líder del proyecto.

3.1.4.2 Etapa de Diseño. En esta etapa las actividades se enfocan en la realización de diseños partiendo de los requisitos establecidos en la etapa de análisis y para ello se tuvo en cuenta el STD 1028-1997 y STD IEEE 730-2002 que asegura la calidad del software.

El equipo de desarrollo es el encargado de suministrar los diseños tales como prototipos iniciales, diagramas de entidad relación y casos de uso que deberán ser asociados a un respectivo requisito anteriormente aprobado. Al culminar el diseño, se delega al equipo de testing temprano la **Revisión del Diseño**, etapa donde se verifica que realmente cada caso de uso cumpla con el requisito asignado. Esta etapa se realizará tantas veces sea necesario hasta corregir los defectos detectados.

Una vez se hayan corregido los defectos detectados y se tengan definidos los diseños se realizará el plan de pruebas de integración en la **Revisión de Diseño**, el cual es realizado por el equipo de testing temprano y debe ser validado por el Líder del proyecto.

3.1.4.3 Etapa de Diseño Detallado. En esta etapa las actividades se enfocan en la realización de diseños más detallados para definir su arquitectura y estructura, para ello se tuvo en cuenta el STD 1028-1997 y STD IEEE 730-2002 que asegura la calidad del software.

El equipo de desarrollo es el encargado de definir la arquitectura y tecnología a implementar y suministrar los diseños detallados tales como, diagramas de clases y diagrama de actividades.

Al culminar el diseño detallado, se delega al equipo de testing temprano la **Revisión del Diseño Detallado**. Ellos verificarán la arquitectura y estructura planteada por el equipo de desarrollo. Esta etapa se realizará tantas veces sea necesario hasta obtener diseños detallados más eficientes posibles.

Una vez se hayan obtenido los diseños detallados más eficientes y partiendo de las actividades generadas en el diagrama de actividades se realizará el plan de pruebas unitarias en la **Revisión de Diseño Detallado**, el cual es realizado por el equipo de testing temprano y debe ser validado por el Líder del proyecto.

3.1.4.4 Etapa de Pruebas. En esta etapa se tienen en cuenta todas las etapas de pruebas tradicionales las cuales en este proyecto son **Pruebas Unitarias, Pruebas de Integración y Pruebas de aceptación.**

Dichas pruebas se realizarán partiendo del plan de pruebas generado por el equipo de testing temprano en las etapas de revisión de diseño detallado, revisión de diseño y revisión de análisis respectivamente,

El equipo de calidad es el encargado de ejecutar dichas pruebas y mostrar los resultados obtenidos, en caso de no obtener una prueba satisfactoria deberá pasar a la etapa de **Depuración y Cambios**.

4 DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS

Este capítulo tiene como propósito presentar el Documento de Especificación de Requisitos para implementar el prototipo web que permitirá gestionar el marco metodológico propuesto.

Se presentan los requerimientos funcionales, así como los diagramas de caso de uso elaborados.

4.1 REQUERIMIENTOS DEL SISTEMA

SoftV&V un sitio web que tiene como objetivo generar proyectos de creación de software de una manera más eficiente por medio de un marco metodológico que aumenta la calidad del producto y reduce costos realizando pruebas tempranas a lo largo de todo el ciclo de vida.

Las principales funcionalidades que se planean para SoftV&V son:

- Permitir a todo el equipo de trabajo tener almacenada toda la información del proyecto a desarrollar
- Permite que el líder del proyecto tenga un mejor control de proyecto.
- Permite al líder y al equipo de desarrollo gestionar el acceso de los demás equipos de trabajo.
- Permite la interacción de diferentes equipos de trabajo en un solo lugar sin afectar el desarrollo de otro, aumentando así el dialogo entre ellos y el cliente
- Mantiene la trazabilidad necesaria para la implementación de las pruebas
- Aumenta la detección de defectos en etapas tempranas
- Reduce costos de producción a largo plazo
- Aumenta la calidad de los proyectos.
- Permite a todo el equipo de trabajo ver su estado y avance del proyecto
- Permite tener la información actualizada par un mejor desempeño.

4.2 CASOS DE USO ASOCIADOS

A continuación, se muestra los diagramas de los casos de uso más representativos.

Figura 7 Registro y Autenticación

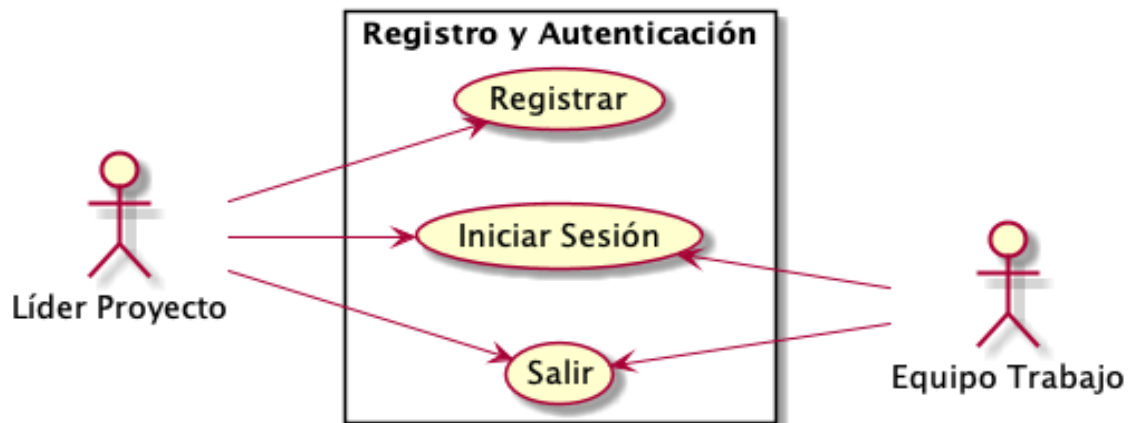


Tabla 2 Caso de uso: iniciar sesión

Nombre del Caso de Uso:	Iniciar Sesión
Característica:	Permite que solo los usuarios ya registrados puedan ingresar a SoftV&V
Descripción:	Esta funcionalidad permite que los usuarios ya registrados en SoftV&V hagan uso adecuado del software y sus características dependiendo del rol asignado.
Precondición:	Estar Registrado y almacenado en la base de datos de SoftV&V
Postcondición:	El ingreso satisfactorio a SoftV&V
Actores:	Líder del proyecto, Equipos de trabajo

Tabla 3 Caso de uso: Registrarse

Nombre del Caso de Uso:	Registrar
Característica:	Permite a los usuarios tener acceso a los servicios de SoftV&V
Descripción:	Esta funcionalidad permite que los líderes nuevos de proyectos software tengan acceso a SoftV&V
Precondición:	No estar Registrado en SoftV&V
Postcondición:	El registro satisfactorio y almacenamiento en la base de datos de SoftV&V
Actores:	Líder del proyecto.

Tabla 4 Caso de uso: iniciar sesión

Nombre del Caso de Uso:	Salir
Característica:	Permite que solo los usuarios terminen su sesión en SoftV&V
Descripción:	Esta funcionalidad permite que los usuarios, que han iniciado sesión en la plataforma de SoftV&V, salgan del aplicativo.
Precondición:	Tener sesión iniciada y vigente en SoftV&V
Postcondición:	La salida satisfactoria de SoftV&V
Actores:	Líder del proyecto, Equipos de trabajo

Figura 8 Caso de uso: Gestión de Proyectos

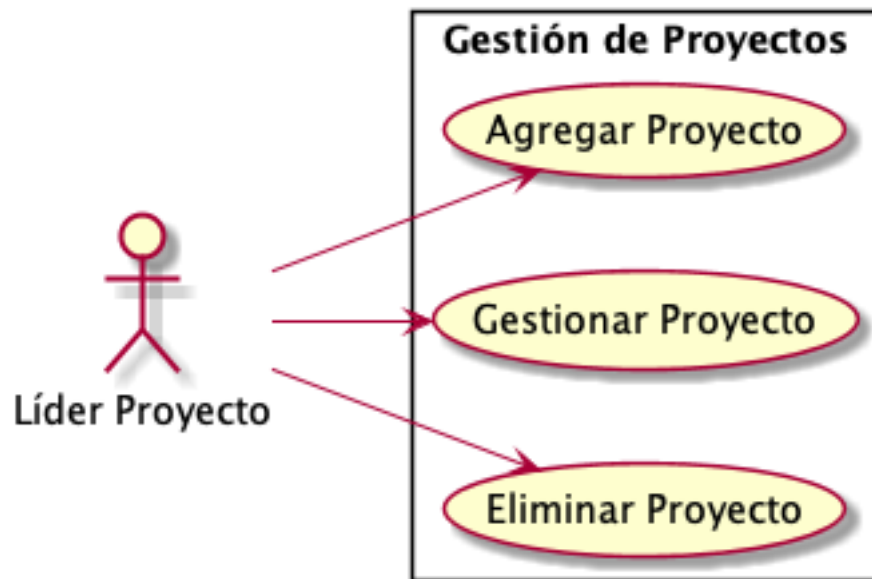


Tabla 5 Caso de uso: Crear proyecto

Nombre del Caso de Uso:	Crear Proyecto
Característica:	Crear nuevos proyectos
Descripción:	Esta funcionalidad permite crear proyectos nuevos de trabajo de SoftV&V
Precondición:	Estar en una sesión activa
Postcondición:	La creación del proyecto satisfactorio.
Actores:	Líder del proyecto.

Tabla 6 Caso de uso: Gestionar proyecto

Nombre del Caso de Uso:	Gestionar proyecto
Característica:	Permitir Gestionar el proyecto.
Descripción:	Esta funcionalidad permite que el líder del proyecto gestione el proyecto sin ninguna restricción.
Precondición:	Tener creado un proyecto
Postcondición:	La gestión del proyecto satisfactoriamente.
Actores:	Líder del proyecto.

Tabla 7 Caso de uso: Eliminar proyecto

Nombre del Caso de Uso:	Eliminar proyecto
Característica:	Permitir Eliminar el proyecto.
Descripción:	Esta funcionalidad permite que el líder del proyecto elimine el proyecto en su totalidad en caso de ya no ser requerido.
Precondición:	Tener creado un proyecto
Postcondición:	La eliminación satisfactoria del registro del proyecto en SoftV&V.
Actores:	Líder del proyecto.

Figura 9 Registro de usuarios

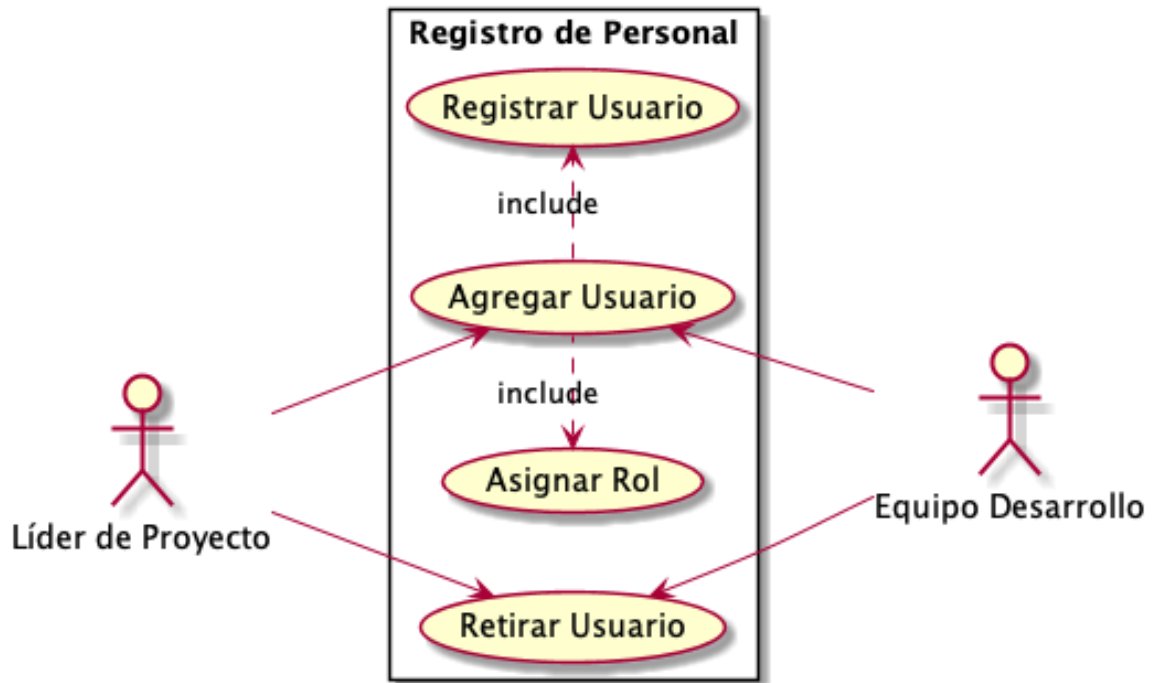


Tabla 8 Caso de uso: Agregar Usuario

Nombre del Caso de Uso:	Agregar Usuario
Característica:	Agrega usuarios previamente registrados o nuevos
Descripción:	Permite agregar usuarios nuevos o previamente registrados al proyecto que se gestiona y asignarles un rol SoftV&V
Precondición:	Estar gestionando proyecto o estar en la etapa de Análisis
Postcondición:	Usuarios agregados satisfactoriamente al proyecto.
Actores:	Líder del proyecto, Equipo de Desarrollo

Tabla 9 Caso de uso: Asignar Rol

Nombre del Caso de Uso:	Asignar Rol
Característica:	Permitir asignar rol a un usuario registrado.
Descripción:	Asigna un rol con sus respectivos permisos a un usuario registrado en la plataforma Soft V&V
Precondición:	Estar registrado en Soft V&V
Postcondición:	Rol asignado satisfactoriamente.
Actores:	Líder del proyecto, Equipo de Desarrollo

Tabla 10 Caso de uso: Eliminar proyecto

Nombre del Caso de Uso:	Retirar Usuario
Característica:	Retira un usuario de SoftV&V de un proyecto
Descripción:	Esta funcionalidad permite que un usuario sea retirado del proyecto que se está gestionando actualmente.
Precondición:	Tener usuarios asignados al proyecto
Postcondición:	El retiro satisfactorio del usuario seleccionado del proyecto en SoftV&V.
Actores:	Líder del proyecto, Equipo de Desarrollo

5 DISEÑO DE SOFTV&V

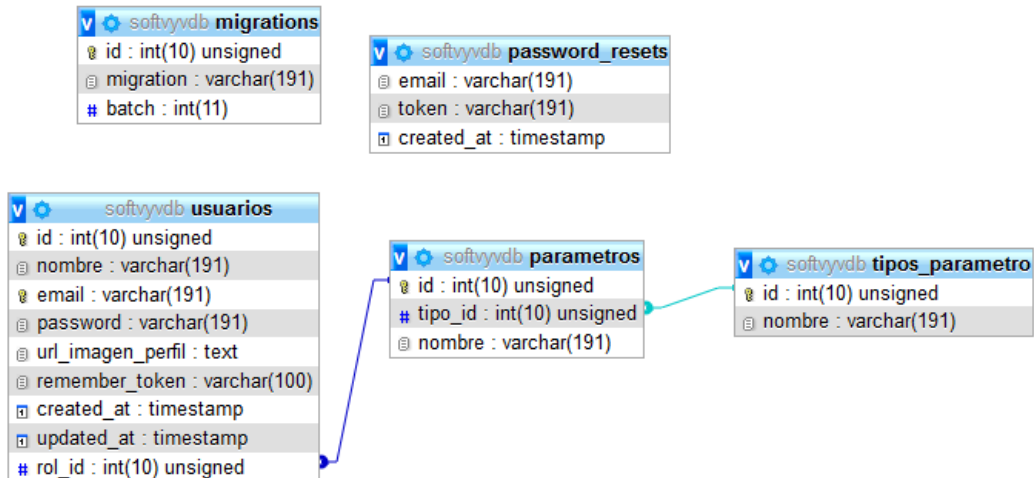
Este capítulo documenta el diseño elaborado para SoftV&V. La base de datos fue diseñada primero en el código de migraciones del framework seleccionado para el desarrollo, pero teniendo en cuenta siempre primero la base de datos seleccionada para su uso.

El framework seleccionado, Laravel, tuvo una influencia en el diseño de SoftV&V ya que este sugiere seguir una arquitectura MVC para el desarrollo, sin embargo, no se está atado a seguir dicha arquitectura para el desarrollo final.

5.1 DIAGRAMAS MER

5.1.1 Tablas del Sistema. En la Figura 10 se observan las tablas básicas del sistema. Tres de ellas son agregadas por Laravel, mientras que las de parámetros fueron agregadas en el diseño y posterior desarrollo.

Figura 10 Tablas del Sistema



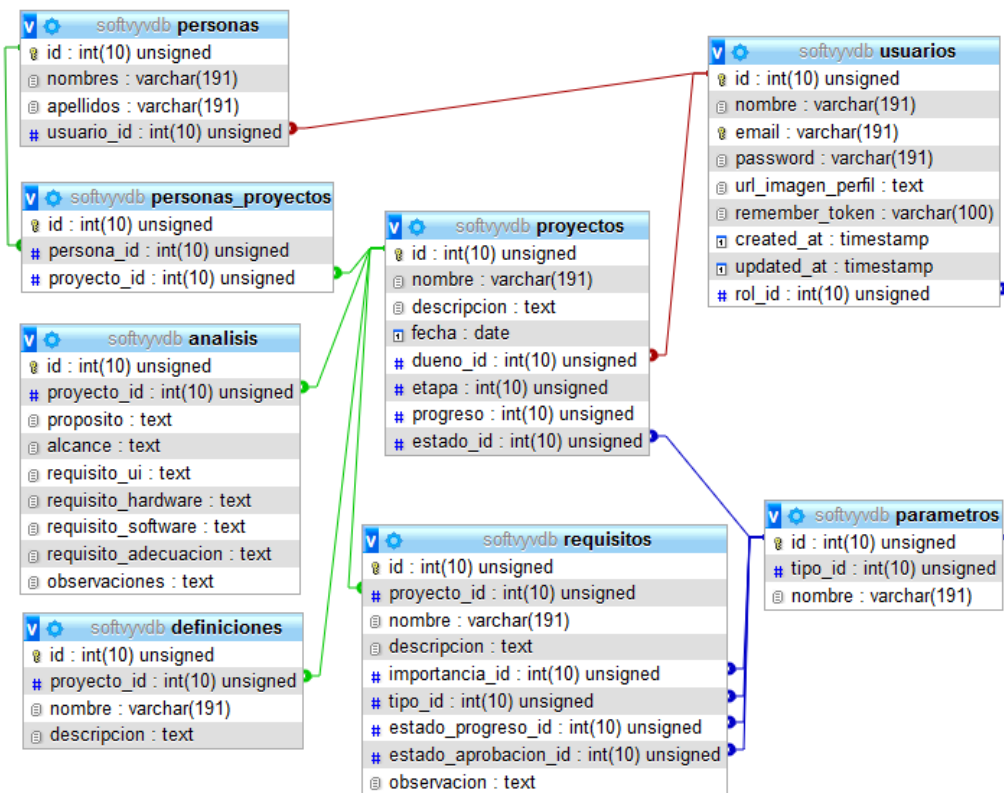
Al usar Laravel, más concretamente, EloquentORM, la tabla *migrations* se agrega automáticamente y se usa para guardar las migraciones de la base de datos que se han aplicado.

Para manejar la autenticación de usuarios es necesario especificar una tabla de *usuarios* la cual provee Laravel por defecto y está parcialmente parametrizada en la versión 5.4 del framework. Por esta razón, la table presenta columnas con nombres en inglés y español. La tabla *password_resets* es usada por Laravel para guardar qué usuario ha pedido recordatorio de contraseña.

Las tablas *tipos_parametro* y *parametros* son usados en SoftV&V en diversas tablas, usos que serán explicados a medida que se vayan mostrando.

5.1.2 Tablas de Análisis y Revisión de Análisis

Figura 11 Tablas etapa de Análisis



Para la etapa de análisis se usan las tablas mostradas en la Figura 11. Los proyectos tienen asignado un *dueño* que es el usuario quien creó el proyecto. Se usó *usuarios* y no *personas* debido a que Laravel tiene acceso a la información del

usuario identificado en la aplicación en todo momento, además que en el diseño inicial se asumió que las tablas del Sistema son independientes a las del aplicativo final. Los proyectos manejan un *progreso* porcentual que se muestra en el listado, y un valor bandera, *etapa*, la cual se usa para habilitar las etapas del modelo W.

En este diagrama se ve que *parametros* es usada para definir el rol de un usuario en el aplicativo, el estado del proyecto, el tipo del requisito (funcional o no funcional), el nivel de importancia del requisito, y para mostrar si el requisito ha sido aprobado o no. El progreso del requisito es una columna que se maneja internamente para verificar si el requisito se ha completado o no (se completa al aprobar las pruebas de aceptación)

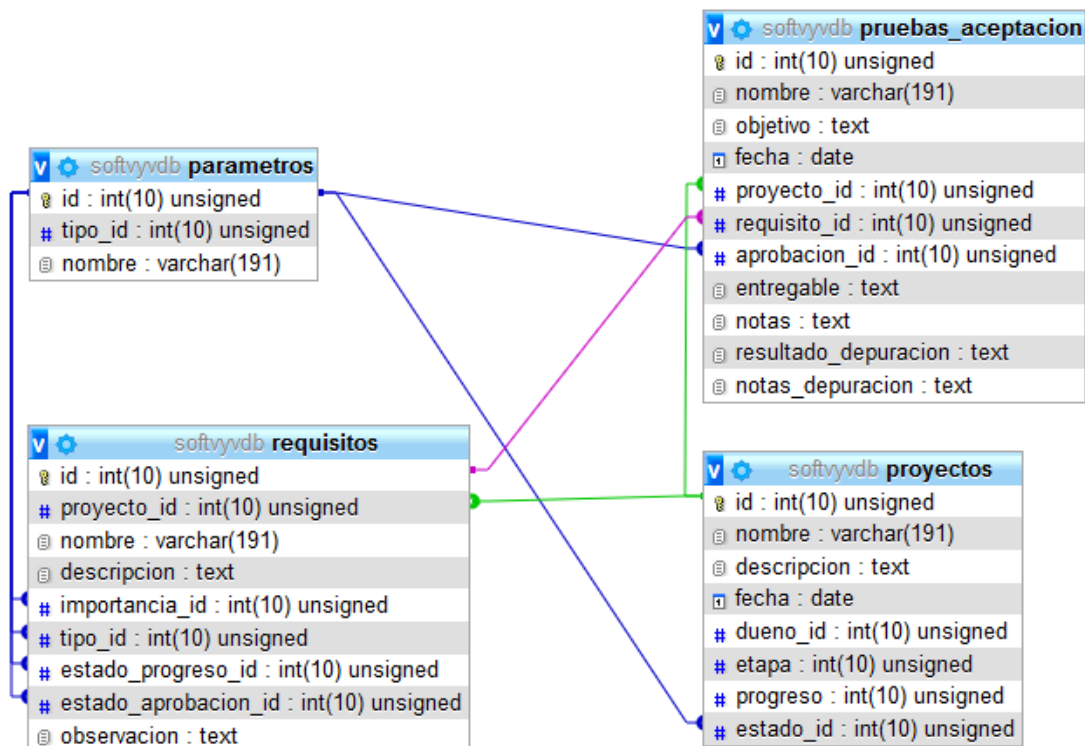
Varias personas pueden ser asignadas a un proyecto, por lo cual se crea la tabla *personas_proyectos*. La tabla *personas* se puede combinar con la de *usuarios*, pero se ha manejado así por claridad en conceptos. Como nota adicional, manejar dicha claridad para futuras implementaciones se puede hacer a nivel de Entidades en Laravel y no necesariamente en la base de datos.

Las otras tablas correspondientes a la etapa de Análisis son *analisis*, *definiciones* y *requisitos*, las cuáles son asignadas con relación uno a uno (en el caso de *analisis*) y uno a muchos con la tabla de *proyectos*.

Las tablas de la revision de análisis se ven en la Figura 12. Es en esta etapa donde se asigna el *estado_aprobacion_id*, el cual se usa para indicar si se aprobó o no el requisito.

La tabla de *pruebas_aceptacion*, se usa para definir el plan de pruebas durante la revisión y se le asigna un único requisito. Las pruebas de aceptación tienen de parámetro su aprobación (aprobado o no aprobado) el cuál se asigna en las etapa de pruebas o en la de Depuración y Cambios. También las pruebas de aceptación son asignadas a un único proyecto (relación uno a uno).

Figura 12 Tablas etapa Revisión de Análisis

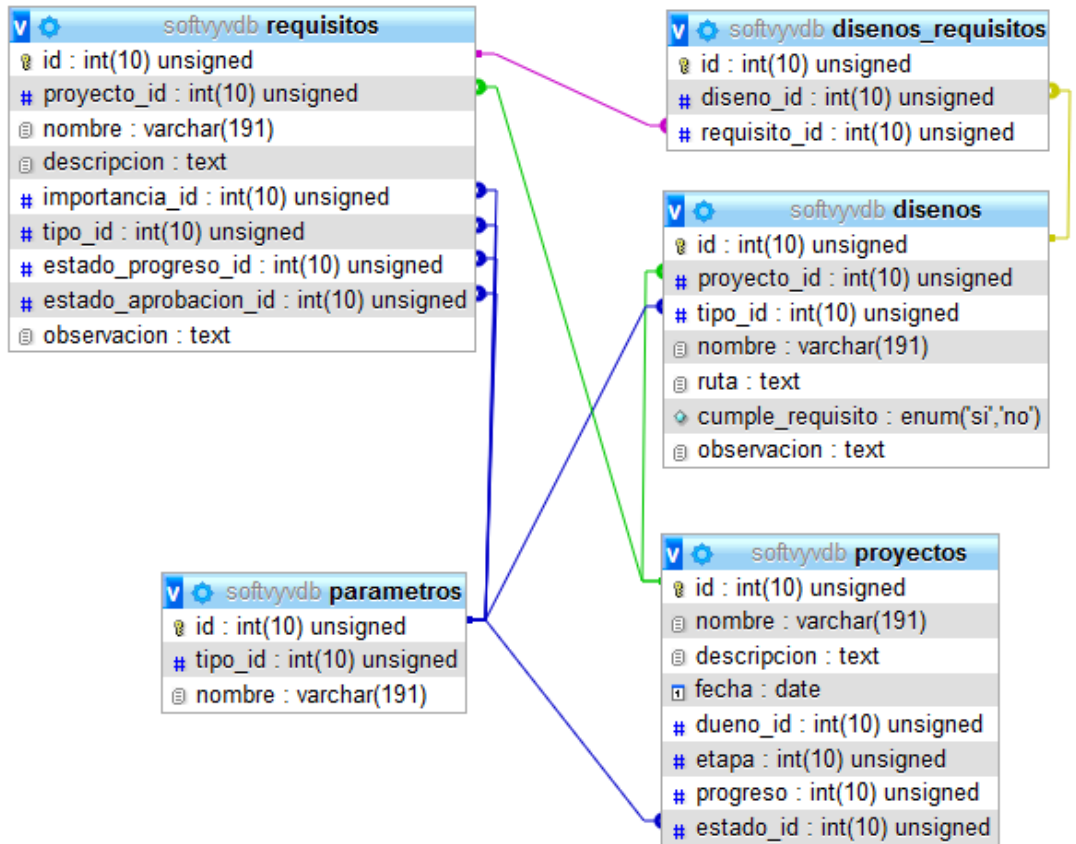


5.1.3 Tablas de Diseño y Revisión Diseño. En la Figura 13 se visualizan las tablas usadas en la etapa de Diseño.

La tabla de *diseños*, tiene una relación uno a muchos con proyectos, ya que se pueden crear varios diseños para un sólo proyecto. En el campo *ruta* se guarda la dirección del archive adjuntado al diseño.

Los tipos de diseño son: diagramas de modelo entidad-relación (como los que se presentan en esta sección), casos de uso y prototipos. Originalmente se había planeado poder asignar más de un requisito a un mismo diseño, por lo cual se creó la tabla *diseños_requisitos* y en una futura mejora se podría implementar esta funcionalidad o refactorizar para eliminar esta relación y simplificarla. En el aplicativo se maneja la relación como si fuera uno a uno. Se tomó esta decisión para forzar a que los requisitos se manejen, en el aplicativo, de forma atómica, es decir, sean muy enfocados y orientados por un principio de responsabilidad única.

Figura 13 Tablas etapa de Diseño

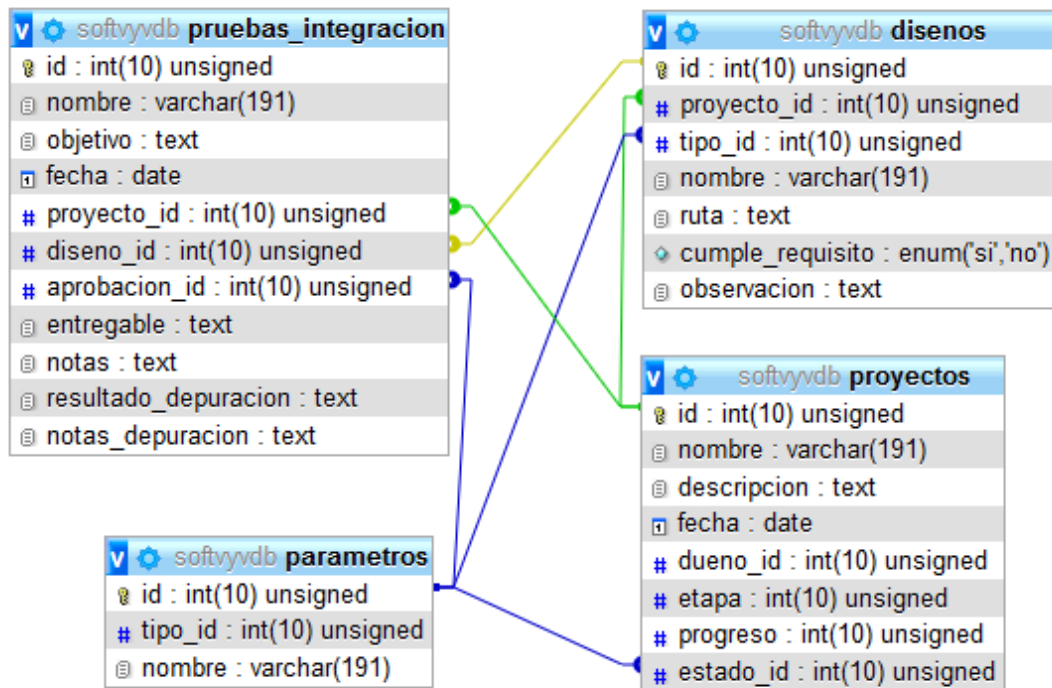


En la Figura 14 se ven las tablas que se usan en la Revisión del Diseño, donde entra la tabla de *pruebas_integracion*, la cual se le asigna un único diseño y se valida en el aplicativo que este sea un Caso de Uso. El parámetro de aprobación se asigna en su etapa de pruebas o en Depuración y Cambios.

Se puede apreciar que la table de Pruebas de Integración es muy similar a la tabla de Pruebas de Aceptación, con la única diferencia siendo que se asocia a un diseño en lugar de un requisito.

El *cumple_requisito* es un campo que se agregó durante el prototipo inicial de SoftV&V, antes de crear la tabla de *parámetros*. En la revisión, normalmente se aprueba o no, pero en el caso del diseño se especifica si cumple con el requisito asignado o no.

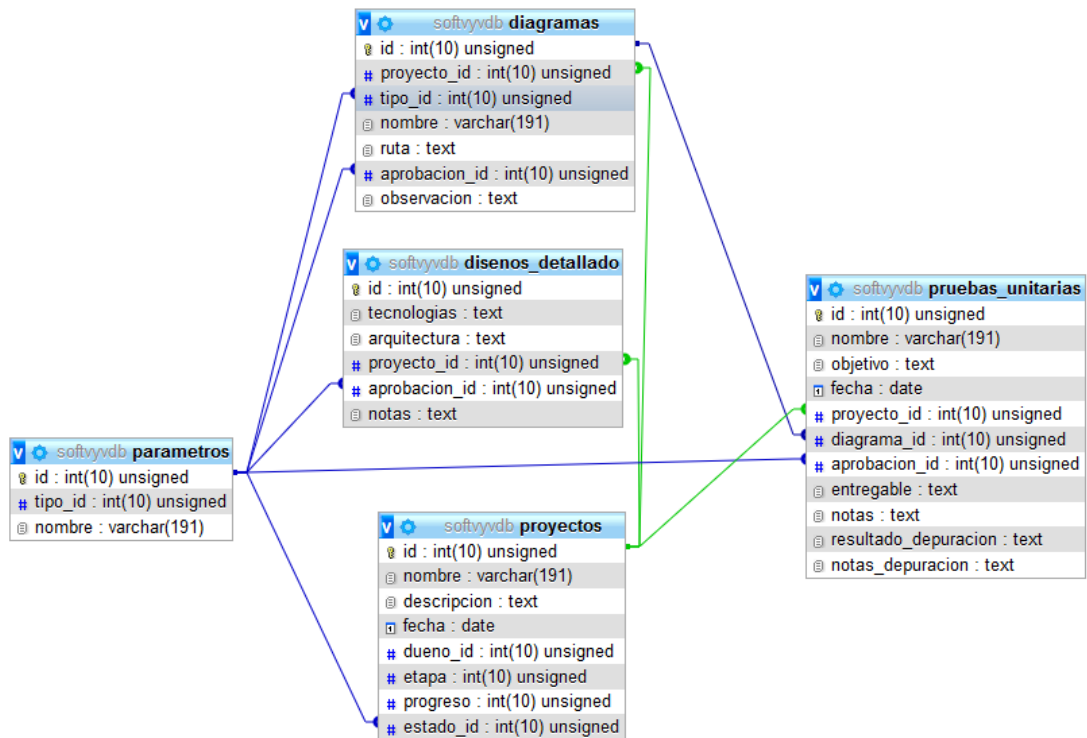
Figura 14 Tablas etapa Revisión Diseño



5.1.4 Tablas de Diseño Detallado y Revisión de Diseño Detallado. Las tablas usadas en la etapa de Diseño Detallado se visualizan en la Figura 15. Están las tablas *disenos_detallado* y *diagramas*. Cada diseño detallado pertenece a un único proyecto y un proyecto sólo tiene un único diseño detallado. Cada proyecto puede tener uno o más diagramas que se diferencian por su parámetro de *tipo* (diagrama de clases o diagrama de actividades)

En la etapa de Revisión del Diseño Detallado, entra la tabla de *pruebas_unitarias*, a la cual se le asigna un único diagrama que se valida en el aplicativo que sea un diagrama de actividad. En esta etapa se aprueba o no el diseño detallado y los diagramas. Las pruebas unitarias se aprueban en su etapa de pruebas o en Depuración y Cambios.

Figura 15 Tablas etapas Diseño Detallado y Revisión de Diseño Detallado



5.1.5 Tablas de Pruebas y Depuración y Cambios En las figuras Figura 12, Figura 14 y Figura 15 se documentan las tablas de Pruebas de Aceptación, Integración y Unitarias, respectivamente, las cuales son las únicas usadas durante las etapas de pruebas y depuración y cambios.

Como se puede apreciar, todas las pruebas son idénticas en la base de datos por excepción de la entidad que se asigna para su verificación. Esto se podría normalizar, pero es mejor reducir el número de Joins posibles a la hora de traer datos, además de que dejar la base de datos de esta manera hace más fácil su razonamiento para realizar mantenimiento. Sin embargo, es un punto para mejorar en una versión futura.

En las respectivas etapas de pruebas, es donde se asignan los campos *notas* y *entregables*. Y en las etapas de Depuración y Cambios se asignan los campos *notas_depuracion* y *resultado_depuracion*. La *fecha* se asigna con la hora y día en la cual se aprueba o no la prueba.

5.2 ARQUITECTURA

Como SoftV&V es un aplicativo web, la arquitectura básica es la de cliente y servidor, sin embargo, el software donde se despliega a pruebas o producción SoftV&V (en este caso Xampp) se encarga de manejar toda la parte de comunicación entre cliente y servidor.

En el aplicativo web como tal, se usa la arquitectura que sugiere Laravel, Modelo – Vista – Controlador o *MVC*.

El flujo de trabajo normal en Laravel consiste en primero diseñar las tablas a agregar a la base de datos o la refactorización de las mismas, luego codificar su respectiva migración, crear el modelo que mapea la tabla a una clase, después el controlador es donde se codifica la lógica de negocio y se maneja la comunicación entre la vista y el modelo, y finalmente la vista, la cual define lo que ve el usuario y cómo interactúa el usuario con el aplicativo.

Los métodos del controlador son entonces mapeados a una ruta la cuál puede tener asignada una vista o un tipo de respuesta en particular. No todos los métodos de un controlador van ligados a una vista, unos netamente se encargan de formatear (A JSON normalmente) los datos consultados de la base de datos.

Cabe destacar que en la vista misma, como es un aplicativo web, también se maneja una arquitectura en este nivel. En este caso se usa JQuery, en el lado del cliente, para manejar una Vista-Modelo asincrónica en el aplicativo web, esto es, se manejan modelos en Javascript para enviar o recibir los datos de la aplicación web mediante rutas tipo REST API.

Las rutas REST API manejan los verbos básicos de Crear, Insertar, Actualizar y Eliminar, y adicional a estos, se maneja *Listar*, para traer más de un registro y mostrarlo en una lista contenida en la Vista.

5.3 INTERFACES PRINCIPALES DE SOFTV&V

A continuación, se muestran las principales interfaces de SoftV&V.

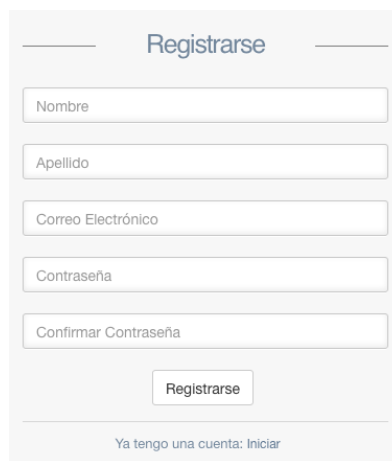
5.3.1 Inicio De Sesión. En esta pantalla es donde los usuarios registrados pueden iniciar sesión en SoftV&V.

Figura 16 Pantalla de Inicio de Sesión



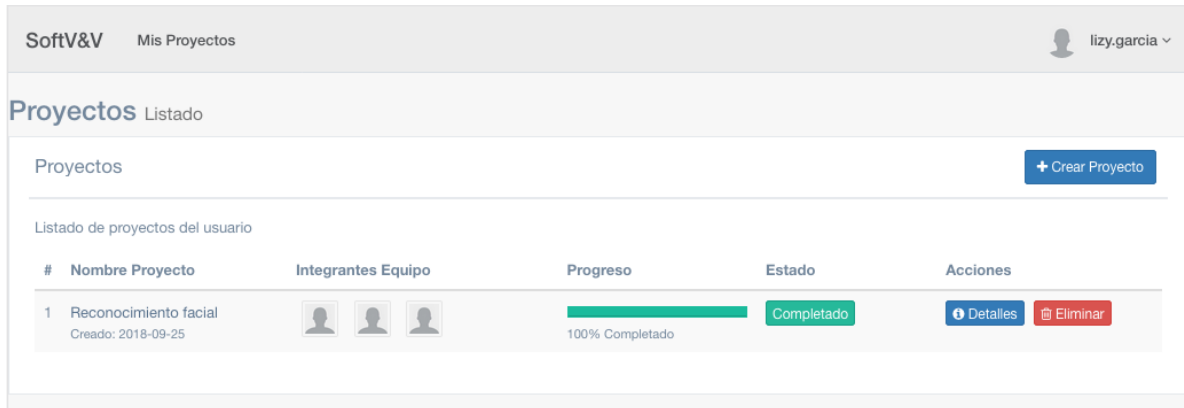
5.3.2 Registro de Nuevo Usuario. En esta pantalla se registran nuevos usuarios. Normalmente usada por líderes de proyecto.

Figura 17 Pantalla de Registro de Usuario



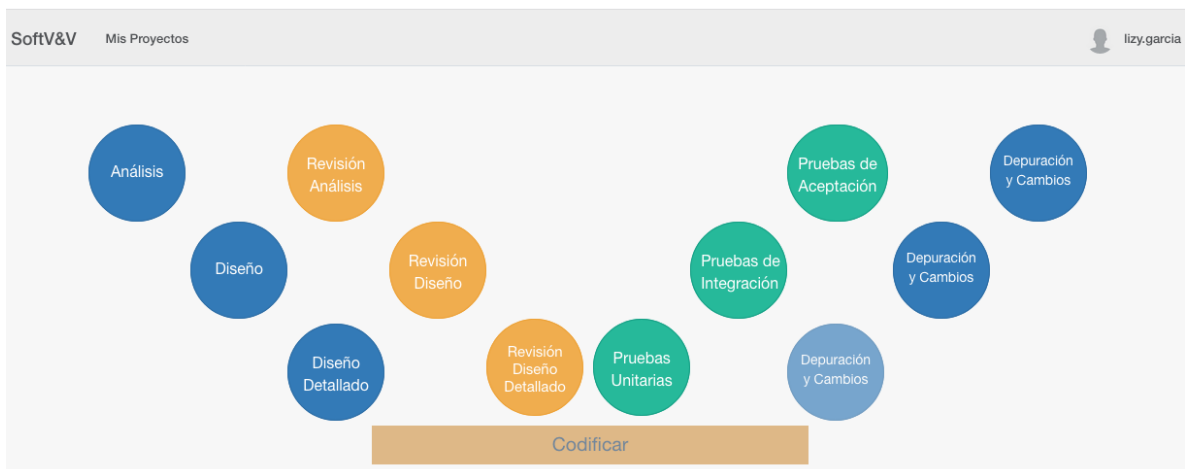
5.3.3 Gestión de Proyectos. En esta pantalla se listan los proyectos a los que se tienen permisos, ya sea como líder del proyecto o como miembro del equipo de trabajo. Aquí se agregan nuevos proyectos, se visualiza el progreso de estos y el Estado o Etapa en el que se encuentran.

Figura 18 Pantalla de Gestión de Proyectos



5.3.4 Modelo W. En esta pantalla se tiene acceso a cada etapa del proyecto, las cuales sólo están habilitadas para el rol que corresponde y si ya se han cumplido los requisitos para habilitar la etapa.

Figura 19 Pantalla de Modelo W



5.3.5 Análisis. En esta pantalla se realiza la etapa de análisis por parte del equipo de desarrollo. En cada pestaña hay una sección con datos para llenar, como las definiciones, los requisitos y el personal que hará parte del equipo de trabajo.

Figura 20 Pantalla Etapa Análisis



SoftV&V Mis Proyectos Modelo W Ayuda lizy.garcia

Análisis

Análisis

En esta etapa se realiza el análisis de requisitos, definiendo el propósito y alcance, añadiendo personal al proyecto, agregando definiciones de términos y formalizando los requisitos del proyecto

General
Definiciones
Personal
Requisitos
Extra

Propósito:
Crear un software que por medio de técnicas biométricas permita la autenticación de usuarios

Alcance:
El objetivo principal es el estudio de las técnicas biométricas para voz y cara y la implementación en un escenario de aula virtual donde contara con la opción de login en la cual se implementara la autenticación biométrica para el acceso.

Guardar Análisis

Atrás

5.3.6 Revisión Análisis. En esta pantalla se realiza la revisión del análisis por parte del equipo de Testing Temprano. En *Revisión* están los requisitos por verificar, en *Plan de Pruebas* se listan las pruebas añadidas al plan y en la última pestaña se listan los requisitos ya aprobados que no han sido asignados a una Prueba.

Figura 21 Pantalla Etapa Revisión de Análisis

SoftV&V Mis Proyectos Modelo W Ayuda lizy.garcia

Revisión de Análisis

Revisión Análisis [+ Agregar Prueba](#)

En esta etapa se diseñarán los modelos de entidad relacion(MER), los casos de uso y los prototipos teniendo en cuenta el cumplimiento de los requisitos aprobados anteriormente.

Revisión | Plan de Pruebas | Requisitos Pendientes por Asignar al Plan

Nombre	Descripción	Tipo de Requisito	Importancia	Estado del Requisito	Observación	Acción
toma de datos personales	a través de un formulario se adquirirán los datos personales de los usuarios	Funcional	Alta	Aprobado	que el formulario este detallado	Editar
toma de datos biométricos	El administrador se manera presencial y mediante la cámara y el micrófono adquirirán las características faciales y de voz de los usuarios	Funcional	Alta	Aprobado	que se tomen una muestra clara de las características faciales y de voz de los usuarios	Editar
registro de usuarios	se almacenara la información de los usuarios en una base de datos que a su vez se conecta con	Funcional	Alta	Aprobado		Editar

5.3.7 Diseño. En esta pantalla se agregan los diseños los cuales cada tipo se lista en una pestaña diferente. Adicionalmente, si faltan requisitos por asignar a un diseño, se visualizan en su propia pestaña.

Figura 22 Pantalla Etapa Diseño

SoftV&V Mis Proyectos Modelo W Ayuda lizy.garcia

Diseño

Diseño [+ Agregar Diseño](#)

En esta etapa se diseñarán los modelos de entidad relacion(MER), los casos de uso y los prototipos teniendo en cuenta el cumplimiento de los requisitos aprobados anteriormente.

Diagramas MER | Casos de Uso | Prototipos | Requisitos pendientes por asignar diseño

Nombre	Requisito	Archivo	Accion
ingreso al aula virtual	registro de usuarios	Descargar	🗑️

5.3.8 Revisión de Diseño. En esta pantalla se revisa cada Diseño por parte del equipo de testing temprano. Adicionalmente se agregan las pruebas para el plan de pruebas y se listan los casos de uso pendientes por asignar prueba.

Figura 23 Pantalla Etapa Revisión Diseño

Revisión Diseño + Agregar Prueba

En esta etapa se diseñaran los modelos de entidad relacion(MER), los casos de uso y los propotipos teniendo en cuenta el cumplimiento de los requisitos aprobados anteriormente.

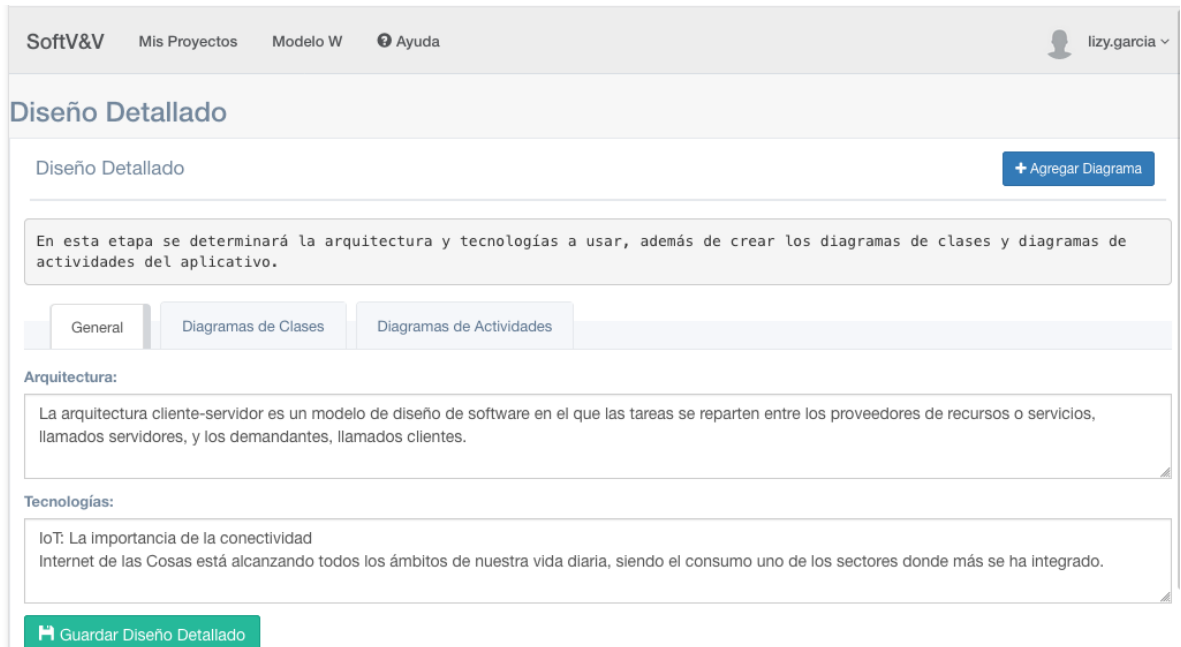
Diagramas MER Casos de Uso Prototipos **Plan de Pruebas** Diseños Pendientes Por Asignar al Plan

Nombre	Requisito	Archivo	Cumple los Requisitos	Observación	Acción
formulario de datos personales	toma de datos personales	Descargar	si	El formulario esta adecuado	Editar
formulario de datos biométricos	toma de datos biométricos	Descargar	si		Editar

5.3.9 Diseño Detallado. Esta etapa la realiza el equipo de desarrollo, y especifican la arquitectura del proyecto, las tecnologías a usar.

Se adjuntan los diagramas de Clases y Actividades que documentan el diseño más detallado del proyecto.

Figura 24 Pantalla Etapa Diseño Detallado



5.3.10 Revisión de Diseño Detallado. Esta etapa es realizada por el equipo de Testing temprano, donde se evalúa la arquitectura, las tecnologías seleccionadas y los diagramas.

Se define el plan de pruebas unitarias. Adicionalmente se listan los diagramas de actividades pendientes por asignar a un plan.

Figura 25 Pantalla Etapa Revisión de Diseño Detallado



5.3.11 Pruebas. Las pantallas de las etapas de pruebas unitarias, integración y aceptación son técnicamente iguales por excepción de la primera columna, la cual muestra el diagrama de actividad, el caso de uso y requisito asignado a la prueba respectivamente.

El equipo de calidad ejecuta las pruebas y documenta los resultados. Se encarga de adjuntar los soportes y de registrar en el sistema si la prueba pasó o no.

Figura 26 Pantalla Etapa Pruebas

Pruebas Unitarias

En esta etapa se realizan las pruebas unitarias verificando que se cumplan. Se adjunta el formato lleno del plan de pruebas o el resultado arrojado por la herramienta usada.

D. Actividad	Nombre	Estado	Fecha Estado	Entregable	Observaciones	Acción
Actividad de registro de usuario	Registro de usuario	Aprobado	2018-09-25	Descargar	paso todas las pruebas	Editar

5.3.12 Depuración y Cambios. Al igual que con la etapa de pruebas, las pantallas de Depuración y Cambios son exactamente iguales por excepción a la primera columna, la cual difiere en el ítem de verificación asociado a la prueba.

El equipo de desarrollo se encarga de la realización de la depuración y cambios y documenta los resultados de ejecutar las pruebas nuevamente tras ejecutar la depuración y cambios. Finalmente, el equipo de calidad da el visto bueno para aprobar los cambios.

Figura 27 Pantalla Etapa Depuración y Cambios

SoftV&V Mis Proyectos Modelo W  Ayuda  lizy.garcia ▾

Depuración y Cambios

Pruebas de Integración

En esta etapa se realizan la depuración y cambios con el fin de que todas las pruebas de integración se cumplan. Se adjunta el formato lleno del plan de pruebas o el resultado arrojado por la herramienta usada.

Casos de Uso	Nombre	Estado	Resultados iniciales	Resultados depuración	Acción
ingreso al aula virtual	verificación de usuario	Aprobado en 2018-09-25	Descargar Observaciones: no paso las prueba	Descargar Observaciones: se arreglo el error y la prueba paso satisfactoriamente	Editar

6 PRUEBAS

En este capítulo se realizan las pruebas funcionales que verifican la funcionalidad correcta de SoftV&V.

6.1 PRUEBAS FUNCIONALES

Con estas pruebas se hizo un proceso de control de calidad a SoftV&V, el cual se iba desarrollando con una metodología iterativa, en la cual se inició con un prototipo que se iba mejorando cada vez más.

Las pruebas documentadas a continuación, son las últimas realizadas para entregar SoftV&V como un ejemplo de aplicación del marco metodológico de pruebas por medio de la validación y verificación.

Las evidencias de los resultados se pueden ver en el numeral 5.3, donde se puede ver un proyecto correctamente ingresado por un usuario.

Tabla 11 Pruebas Pantalla de Ingreso

Caso de Prueba	Resultado
Realiza el registro de un nuevo usuario	Registro exitoso de un nuevo usuario en la plataforma.
Realiza el inicio de sesión de un usuario	Se inicia sesión exitosamente en la plataforma web con las credenciales asignadas a cada usuario.
Redirige a pantalla de inicio de sesión si accede a un link del proyecto sin tener sesión activa	La plataforma redirige a la pantalla de inicio de sesión tras acceder mediante links de diferentes secciones de la plataforma sin haber iniciado sesión

Tabla 12 Pruebas Pantalla Gestión Proyectos

Caso de Prueba	Resultado
Crea un proyecto en la lista	Se crean proyectos nuevos de manera exitosa.
Lista proyectos en los cuales está agregado como miembro de equipo	Lista de manera correcta los proyectos a los que está asignado o en los que figura como dueño del proyecto.

Tabla 13 Pruebas Pantalla Modelo W

Caso de Prueba	Resultado
Bloquea el acceso a etapas donde no se han cumplido las condiciones de habilitación	La plataforma muestra las etapas deshabilitadas cuando no se ha completado la etapa actual.

Tabla 14 Pruebas Etapa Análisis

Caso de Prueba	Resultado
Guarda el propósito y alcance del Proyecto	Se guarda exitosamente en la base de datos el alcance y propósito del proyecto.
Guarda definiciones pertinentes al proyecto	Se guardan exitosamente en la base de datos los registros de definiciones para el proyecto.
Se pueden eliminar definiciones del proyecto	Se eliminan exitosamente en la base de datos los registros de definiciones del proyecto.
Permite agregar personal al proyecto	El sistema permite agregar correctamente personal al proyecto. Crea el usuario cuando es nueva persona y aparece con los datos existentes cuando es usuario ya registrado en la plataforma.
Permite retirar personal del proyecto	El sistema permite retirar personal del proyecto de manera correcta sin eliminar su registro de usuario.
Permite agregar requisitos al proyecto	El sistema agrega correctamente a la base de datos los registros de requisitos asignados al proyecto
Permite eliminar requisitos del proyecto	El sistema permite eliminar con satisfacción los requisitos seleccionados del proyecto de la base datos.
Permite guardar requisitos adicionales pertinentes al análisis	El sistema permite guardar los requisitos adicionales especificados para el proyecto.
Habilita revisión de Análisis al completar la etapa	El sistema habilita la siguiente etapa cuando se define el propósito y alcance, hay por lo menos un requisito y se ha añadido personal al proyecto.

Tabla 15 Pruebas Etapa Revisión de Requisitos

Caso de Prueba	Resultado
Permite asignar nivel de importancia a los requisitos	El sistema permite asignar uno de los tres niveles de importancia para un requisito.
Permite aprobar o no aprobar los requisitos	El sistema permite marcar como Aprobado o No Aprobado un requisito, adjuntado notas de observación adicionales.
Lista los requisitos aprobados sin asignar a pruebas	Al aprobar requisitos, se va actualizando la lista de requisitos aprobados pendientes de asignación de forma correcta.
Permite agregar pruebas de aceptación asignando un requisito aprobado	El sistema permite agregar pruebas de aceptación asignando un requisito a estos de manera correcta.
Permite eliminar pruebas de aceptación	El sistema permite eliminar pruebas de aceptación del listado de manera correcta.
Actualiza la lista de requisitos aprobados pendientes por asignar a medida que agrega o elimina pruebas	El sistema actualiza de forma dinámica la lista de requisitos aprobados pendientes de asignación, removiendo requisitos del listado al agregar pruebas o agregando el requisito al eliminar pruebas.
Habilita la etapa de diseño al completar la revisión del análisis	El sistema habilita la siguiente etapa tras tener todos los requisitos aprobados con su respectiva prueba de aceptación.

Tabla 16 Pruebas Etapa Diseño

Caso de Prueba	Resultado
Permite agregar diseños según el tipo	Los diseños se agregan correctamente tras asignarles un requisito y adjuntando el archivo correspondiente.
Lista cada tipo de diseño por separado	El sistema lista correctamente cada tipo de diseño en su respectiva lista.
Permite eliminar diseños	El sistema permite eliminar diseños previamente agregados.
Lista los requisitos sin asignar a un diseño	El sistema lista correctamente los requisitos pendientes por asignar diseño.
Actualiza la lista de requisitos sin asignar según se agregan o quitan diseños	El sistema actualiza dinámicamente la lista de requisitos pendientes de manera correcta.
Habilita la etapa de Revisión de Diseño tras completar el Diseño	El sistema habilita la siguiente etapa tras añadir al menos un caso de uso y asignar todos los requisitos a un diseño.

Tabla 17 Pruebas Etapa Revisión Diseño

Caso de Prueba	Resultado
Permite especificar si el diseño cumple o no el requisito	El sistema permite especificar si el diseño cumple o no el requisito de manera correcta.
Permite añadir notas de observación al diseño	El sistema permite correctamente añadir notas de observación al diseño.
Lista los casos de uso aprobados sin asignar prueba de integración	El sistema lista correctamente los casos de uso aprobados pendientes de asignación.
Permite añadir pruebas de integración asignando casos de uso aprobados	Se agregan pruebas de integración con asignación de casos de usos de manera correcta.
Permite eliminar pruebas de integración añadidas	El sistema permite eliminar prueba de integración satisfactoriamente.
Actualiza la lista de casos de usos aprobados pendientes de asignación a medida que añade o elimina pruebas de integración	El sistema actualiza dinámicamente la lista de casos de uso pendientes. Los retira de la lista al asignarlos a una prueba y los reintegra al eliminar la prueba.
Habilita la etapa de Diseño Detallado tras completar la Revisión de Diseño	La siguiente etapa se habilita de manera correcta cuando se aprueban todos los diseños y se asignan todos los casos de uso a una prueba.

Tabla 18 Pruebas Etapa Diseño Detallado

Caso de Prueba	Resultado
Permite guardar la arquitectura y tecnologías especificadas en el diseño detallado	El sistema guarda satisfactoriamente la arquitectura y tecnologías especificadas en el diseño detallado.
Permite añadir diagramas de clase y actividades	Se agregan diagramas de clase y actividades de forma correcta.
Permite eliminar diagramas de clase o actividades	Se eliminan los diagramas de clase o actividad seleccionado de forma correcta.
Lista los diagramas separados por tipo	Los diagramas de clase y actividades se listan correctamente en sus secciones respectivas.
Habilita la etapa de Revisión Diseño Detallado tras completar la etapa de Diseño Detallado	El sistema habilita la siguiente etapa de manera correcta tras definir la arquitectura y tecnologías y al agregar al menos un diagrama de cada tipo.

Tabla 19 Pruebas Etapa Revisión Diseño Detallado

Caso de Prueba	Resultado
Permite aprobar o no la arquitectura y tecnologías especificadas.	El sistema permite correctamente marcar la arquitectura y tecnologías como aprobadas o no aprobadas.
Permite aprobar o no los Diagramas de clase y de actividades, y añadir sus observaciones respectivas	El sistema permite aprobar o no aprobar los diagramas de clase y de actividades adjuntando sus observaciones respectivas.
Lista los diagramas de actividades aprobados pendientes por asignar plan	El sistema lista correctamente los diagramas de actividades aprobados pendientes por asignar plan.
Permite añadir pruebas unitarias asignado un diagrama de actividades	Se agregan exitosamente pruebas unitarias con diagramas de actividades asignado.
Permite eliminar pruebas unitarias	El sistema permite eliminar correctamente pruebas unitarias de la lista.
Actualiza la lista de diagramas pendientes por asignar a medida que añade o elimina pruebas	El sistema actualiza correctamente la lista de diagramas pendientes, quitándolos al asignar pruebas y reintegrándolos al remover la prueba.
Habilita la etapa de Pruebas Unitarias tras completar la revisión de diseño detallado	Se prosigue a la siguiente etapa de pruebas tras aprobar las tecnologías, la arquitectura, todos los diagramas y asignar todos los diagramas de actividad a una prueba unitaria.

Tabla 20 Pruebas Etapa de Pruebas

Caso de Prueba	Resultado
Permite aprobar o no la prueba, adjuntando entregable de la con su respectiva nota	Se pueden aprobar o no las pruebas adjuntando entregable y notas de forma correcta.
Habilita depuración y cambios si hubo pruebas no aprobadas	Si habilita depuración y cambios cuando hay al menos una prueba no aprobada.
Habilita la siguiente etapa de pruebas si se aprobaron todas las pruebas y no está habilitado depuración y cambios	Se habilita la siguiente etapa de pruebas si se aprueban todas las pruebas y depuración y cambios no se habilitó.
Muestra el proyecto como completado si completo la etapa de pruebas de aceptación con todas las pruebas aprobadas	El sistema muestra un mensaje de Proyecto completado al terminar las pruebas de aceptación con todo aprobado al primer intento.

Tabla 21 Pruebas Etapa Depuración y Cambios

Caso de Prueba	Resultado
Permite aprobar o no la prueba, adjuntando entregable de depuración y cambios con su respectiva nota	El sistema permite aprobar o no la prueba, adjuntando el respectivo entregable y observaciones de depuración y cambios.
Habilita la siguiente etapa de pruebas si se aprueban todas las pruebas en depuración de cambios	El sistema habilita la siguiente etapa de pruebas si se aprueban todas las pruebas durante depuración y cambios.
Muestra el proyecto como completado si todas las pruebas de aceptación figuran como aprobadas en Depuración y Cambios	El sistema muestra un mensaje de proyecto completado si todas las pruebas de aceptación figuran como aprobadas en su etapa de Depuración y Cambios activa.

7 CONCLUSIONES

Se logró plantear una propuesta de un marco metodológico que apoya las buenas prácticas de realizar pruebas que permiten la verificación y validación de un producto software durante todo su ciclo de vida.

La definición del marco metodológico requirió la creación de un conjunto de normas y plantillas, las cuales ayudan con el desarrollado y gestión de tareas relacionadas con las actividades de verificación y validación de productos software, lo que aporta a elevar el nivel de calidad de este.

Se diseñaron módulos de apoyo los procesos de verificación y validación con su respectivo modelo de información, para las etapas de Análisis, Diseño, Diseño Detallado y Pruebas del ciclo de vida de un producto de software.

Se desarrolló de forma iterativa una versión inicial de una aplicación web para la gestión de la información y recursos disponibles para el marco metodológico propuesto, implementando en un ejemplo concreto la metodología propuesta.

Se realizaron pruebas al marco metodológico mediante el aplicativo web desarrollado, lo cual permitió tener un ejemplo concreto de la metodología en acción y cómo mejora la calidad del producto de software al aplicarlo.

Se mejoró la apreciación de las buenas prácticas en ingeniería del software sobre pruebas, verificación y validación tras aplicarlas en la realización del aplicativo web que implementa el marco metodológico.

8 RECOMENDACIONES

Hace falta poner más a prueba con desarrollos concretos y más diversos el marco metodológico, en especial en conjunto a metodologías ágiles las cuales han creado una fama de que la documentación es innecesaria en las etapas de desarrollo de software.

Es posible que sea necesario ajustar el marco metodológico para aumentar su compatibilidad con la realidad de las metodologías ágiles que hoy en día lideran en el desarrollo de software.

Es necesario mejorar la base de datos del aplicativo web, el cual aún presenta herencia de las primeras iteraciones del prototipo. Se recomienda realizar pruebas de estrés y de carga al diseño de la base de datos, ya que con datos concretos se pueden tomar decisiones más acertadas sobre qué tablas normalizar y cuáles hay que desnormalizar o mantener desnormalizadas. Dichas pruebas se salen del alcance del proyecto actual, por lo que podría ser útil para avances del marco metodológico y su implementación en software.

En el desarrollo del aplicativo web se eligió un framework para acelerar el desarrollo del mismo, pero sólo para el back end. Otro avance en el proyecto podría ser usar un framework en el front end, lo cual podría ayudar a simplificar aún más el desarrollo y en especial el mantenimiento de la aplicación.

BIBLIOGRAFÍA

AGURRE. Lizette.,2005 [en línea] Marco Metodológico del proceso de verificación y validación de software para pequeñas y medianas empresas. (recuperado el 20 de agosto de 2018). Disponible en: <https://core.ac.uk/download/pdf/54217032.pdf>

BARRY W. Boehm, Guidelines for Verifying and Validating Software Requirements and Design Specifications, 1984. p. 75.

Bootstrap [En línea] Conjunto de herramientas (recuperado el 20 de agosto de 2018). Disponible en: <https://getbootstrap.com/>

ESTEBAN. Ana.,2012 [en línea] Marco Metodológico para la mejora de las actividades de verificación y validación de productos software. (recuperado el 20 de agosto de 2018). Disponible en: https://e-archivo.uc3m.es/bitstream/handle/10016/15464/tesis_ana_sanz_esteban_2012.pdf

Laravel [En línea] framework de código (recuperado el 20 de agosto de 2018). Disponible en: <https://laravel.com/>

Modelo en V [en línea] (recuperado el 20 de agosto de 2018). Disponible en: <http://www.gerrardconsulting.com/?q=node/531>

Modelo en W de MADEJA [en línea] Marco de Desarrollo de Software de la Junta de Andalucía. (recuperado el 20 de agosto de 2018). Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/verificacion>

PUELLO, O., Modelo de Verificación y Validación Basado en CMMI. *Investigación e Innovación en Ingenierías*, 2013 p. 1.

SOMMERVILLE, Ingeniería del software, 2005. p. 472.

SPILLNER, A., & BREMENN, H., The W-MODEL. Strengthening the bond between development and test. In p. 15-17