

Documentación de la técnica TDD en el marco de una metodología ágil utilizando como caso de estudio el desarrollo de un software de control de proyectos de grados para el programa de ingeniería de sistemas de la UNIAJC

Cristian Leandro Botina Caipe ✉ cristian_botina@hotmail.com

Trabajo de Grado presentado para optar al título de Ingeniero en Sistemas

Asesor: Beatriz Eugenia Marín Ospina Especialista (Esp) en Ingeniería de Software



Institución Universitaria Antonio José Camacho

Facultad de Ingenierías

Ingeniería de Sistemas

Cali - Colombia

2018

Nota de aceptación:

**Aprobado por el Comité de Grado en
cumplimiento de los requisitos exigidos por
la Institución Universitaria Antonio José
Camacho para optar al título de Ingeniero
de Sistemas**

Jurado

Jurado

Santiago de Cali, 20 de septiembre de 2017

Dedicatoria

Quiero dedicar este proyecto a mi esposa por tener tanta paciencia y aguantar mi ausencia en este proceso de desarrollo de mi etapa profesional en especial en la ejecución de este proyecto.

Agradecimientos

Quiero agradecer primeramente a Dios por brindarme esa perseverancia y luego quiero agradecer el apoyo y orientación de la directora del proyecto

TABLA DE CONTENIDO

1.	Planteamiento del Problema	14
1.1	Pregunta de Formulación	15
1.2	Pregunta de Sistematización	15
2.	Objetivos.....	16
2.1	Objetivo General.....	16
2.2	Objetivos Específicos	16
3.	Justificación	17
4.	Alcances y limitaciones	18
4.1	Limitaciones	18
5.	Marco Referencial	19
5.1	Antecedentes.....	19
5.2	Marco Conceptual.....	22
5.2.1	Test Driven Development (TDD).	22
5.2.2	Acceptance Test Driven Development (ATDD)	22
5.2.3	Developer TDD	23
5.3	Marco Teórico	23
5.3.1	Fases TDD.....	23
5.3.2	ATDD (Acceptance Test Driven Development)	24
5.3.3	TDD (Test Driven Development) paso a paso	25
5.3.4	SCRUM.....	26
5.3.4.1	Fases de SCRUM	27
6.	Desarrollo	29
6.1	Inicio.....	30

6.1.1	Crear la visión del proyecto.	30
6.1.2	Requerimientos Funcionales	30
6.1.3	Requerimientos No Funcionales	32
6.1.4	Product Backlog	33
6.1.5	Priorización del Product Backlog.....	35
6.1.6	Prioritized Product Backlog	35
6.2	Planear y Estimar.....	38
6.2.1	Historias de usuario	38
6.2.2	Aprobar y estimar historias de usuarios.	43
6.2.2.1	Sprint 1	45
6.2.2.2	Sprint 2	45
6.2.2.3	Sprint 3	45
6.2.2.4	Sprint 4	45
6.2.2.5	Sprint 5	46
6.2.2.6	Sprint 6	46
6.2.2.7	Sprint 7	46
6.2.2.8	Sprint 8	46
6.3	Implementar.....	46
6.3.1	Aplicación de la técnica TDD	46
6.3.2	Sprint 1	58
6.3.2.1	Sprint Backlog	58
6.3.2.2	Log de Impedimentos	59
6.3.2.3	Riesgos Identificados y Mitigados	59
6.3.3	Sprint 2	60
6.3.3.1	Sprint Backlog	60

6.3.3.2	Log de Impedimentos	61
6.3.3.3	Riesgos Identificados y Mitigados	61
6.3.4	Sprint 3	62
6.3.4.1	Sprint Backlog	62
6.3.4.2	Log de Impedimentos	64
6.3.4.3	Riesgos Identificados y Mitigados	64
6.3.5	Sprint 4	64
6.3.5.1	Sprint Backlog	64
6.3.5.2	Log de Impedimentos	65
6.3.5.3	Riesgos Identificados y Mitigados	65
6.3.6	Sprint 5	66
6.3.6.1	Sprint Backlog	66
6.3.6.2	Log de Impedimentos	67
6.3.6.3	Riesgos Identificados y Mitigados	67
6.3.7	Sprint 6	67
6.3.7.1	Sprint Backlog	67
6.3.7.2	Log de Impedimentos	68
6.3.7.3	Riesgos Identificados y Mitigados	68
6.3.8	Sprint 7	68
6.3.8.1	Sprint Backlog	69
6.3.8.2	Log de Impedimentos	69
6.3.8.3	Riesgos Identificados y Mitigados	69
6.3.9	Sprint 8	69
6.3.9.1	Sprint Backlog	69
6.3.9.2	Log de Impedimentos	71

6.3.9.3	Riesgos Identificados y Mitigados	71
6.4	Revisión y Retrospectiva	71
6.4.1	Revisión del sprint.....	71
6.4.1.1	Pruebas.	72
6.4.2	Retrospectiva del sprint.....	77
6.5	Lanzamiento	77
6.5.1	Encuesta	77
6.6	Ventajas y Desventajas de Aplicar La Técnica TDD En La Metodología Scrum	82
7.	Diseño de Software.....	85
7.1	Arquitectura	85
7.1.1	Vista lógica y desarrollo.....	86
7.1.1.1	Diagrama De Clases	86
7.1.2	Vista de proceso	87
7.1.2.1	Vista de secuencia	87
7.1.2.2	Diagrama de actividad	88
7.1.3	Vista física.....	89
7.1.3.1	Diagrama de despliegue	89
7.2	Base de Datos	89
7.2.1	Modelo Entidad Relación (MER)	90
8.	Resultados.....	91
9.	Conclusiones.....	91
10.	Recomendaciones y Trabajos Futuros	94
11.	Referencias Bibliográficas.....	95
12.	Anexos	97
12.1	Anexo 1	97

12.2	Anexo 2.....	97
12.3	Anexo 3.....	97
12.4	Anexo 4.....	97
12.5	Carpeta Anexo 5	97
12.6	Anexo 6.....	97
12.7	Anexo 7.....	98
12.8	Anexo 8.....	98
12.9	Anexo 9.....	98
12.10	Anexo 10.....	98

LISTA DE TABLAS

Tabla 1 <i>Requerimientos Funcionales</i>	30
Tabla 2 <i>Requerimientos No Funcionales</i>	32
Tabla 3 <i>Product Backlog</i>	33
Tabla 4 <i>Prioritized Product Backlog</i>	35
Tabla 5 <i>Lista TO-DO de pruebas</i>	48
Tabla 6 <i>Formato de aplicación de la técnica TDD 1</i>	49
Tabla 7 <i>Formato de aplicación de la técnica TDD 2</i>	51
Tabla 8 <i>Sprint Backlog Sprint 1</i>	59
Tabla 9 <i>Sprint Backlog Sprint 2</i>	60
Tabla 10 <i>Sprint Backlog Sprint 3</i>	62
Tabla 11 <i>Sprint Backlog Sprint 4</i>	64
Tabla 12 <i>Sprint Backlog Sprint 5</i>	66
Tabla 13 <i>Sprint Backlog Sprint 6</i>	67
Tabla 14 <i>Sprint Backlog Sprint 7</i>	69
Tabla 15 <i>Sprint Backlog Sprint 8</i>	70
Tabla 16 <i>Listado de Casos de Pruebas</i>	72
Tabla 17 <i>Comparación SCRUM con Hibrido Scrum + TDD</i>	82

LISTA DE FIGURAS

Figura 1 Ciclo de vida TDD	23
Figura 2 Flujo de SCRUM	27
Figura 3 Diagrama de SCRUM con TDD	29
Figura 4 Historias de Usuario	39
Figura 5 Aprobación y Estimación de Historias de Usuario	44
Figura 6 Sprint Planning	45
Figura 7 Ejecución de los tests	47
Figura 8 Estructura del Test Unitario	47
Figura 9 Aplicación técnica TDD 1	49
Figura 10 Aplicación técnica TDD 2	49
Figura 11 Aplicación técnica TDD 3	50
Figura 12 Aplicación técnica TDD 4	50
Figura 13 Aplicación técnica TDD 5	50
Figura 14 Aplicación técnica TDD 6	51
Figura 15 Aplicación técnica TDD 7	52
Figura 16 Aplicación técnica TDD 8	53
Figura 17 Aplicación técnica TDD 9	53
Figura 18 Aplicación técnica TDD 10	54
Figura 19 Aplicación técnica TDD 11	54
Figura 20 Aplicación técnica TDD 12	54
Figura 21 Aplicación técnica TDD 13	55
Figura 22 Aplicación técnica TDD 14	55
Figura 23 Aplicación técnica TDD 15	56
Figura 24 Aplicación técnica TDD 16	56
Figura 25 Aplicación técnica TDD 17	57
Figura 26 Aplicación técnica TDD 18	57
Figura 27 Aplicación técnica TDD 19	57
Figura 28 Aplicación técnica TDD 20	58

Figura 29 Aplicación técnica TDD 21	58
Figura 30 Resumen pregunta 1 del cuestionario de aceptación de la aplicación web	
ProyectoPlus	78
Figura 31 Resumen pregunta 2 del cuestionario de aceptación de la aplicación web	
ProyectoPlus	79
Figura 32 Resumen pregunta 3 del cuestionario de aceptación de la aplicación web	
ProyectoPlus	79
Figura 33 Resumen pregunta 4 del cuestionario de aceptación de la aplicación web	
ProyectoPlus	79
Figura 34 Resumen pregunta 5 del cuestionario de aceptación de la aplicación web	
ProyectoPlus	80
Figura 35 Resumen pregunta 6 del cuestionario de aceptación de la aplicación web	
ProyectoPlus	80
Figura 36 Resumen pregunta 7 del cuestionario de aceptación de la aplicación web	
ProyectoPlus	80
Figura 37 Resumen pregunta 8 del cuestionario de aceptación de la aplicación web	
ProyectoPlus	81
Figura 38 Resumen pregunta 9 del cuestionario de aceptación de la aplicación web	
ProyectoPlus	81
Figura 39 Resumen pregunta 10 del cuestionario de aceptación de la aplicación web	
ProyectoPlus	81
Figura 40 Resumen pregunta 11 del cuestionario de aceptación de la aplicación web	
ProyectoPlus	82
Figura 41 Diagrama de Clases	86
Figura 42 Diagrama de Secuencia Asignar Evaluador.....	87
Figura 43 Diagrama de Actividades.....	88
Figura 44 Diagrama de Actividades.....	89
Figura 45 Modelo de Entidad Relación (MER)	90

Resumen

A lo largo de este documento se presenta la experiencia obtenida al realizar un desarrollo de software web para la universidad Antonio José Camacho aplicando la técnica Test Driven Development con los elementos Scrum, encontrando las ventajas y desventajas de la técnica de programación con la experiencia propia y enriquecedora adquirida.

Para lograr esto se propuso trabajar técnica Test Driven Development con la metodología Scrum aplicando las fases de inicio, la planeación y estimación, se hace énfasis en la fase de implementación ya que es donde se aplica la técnica en la programación, analizando cada día un listado de pruebas y aplicando los pasos del ciclo Test Driven Development en ese listado para ir logrando terminar la historia de usuario paso a paso de forma incremental.

Basado en la experiencia obtenida en el desarrollo de este proyecto se puede concluir que la aplicación de la técnica es una mejor alternativa para el desarrollo de software logrando una mayor calidad en producto de software entregado y el programador puede realizar menos trabajo resolviendo errores que se pueden producir a futuro.

Abstract

Throughout this document we present the experience obtained when developing a web software for Antonio José Camacho University applying the Test Driven Development technique with the Scrum elements, finding the advantages and disadvantages of the programming technique with the own experience and acquired enrichment.

To achieve this, it was proposed to work with Test Driven Development technique with the Scrum methodology, applying the phases of initiation, planning and estimation, emphasis is placed on the implementation phase since it is where the technique is applied in the programming, analyzing every day a list of tests and applying the steps of the Test Driven Development cycle in that list to be able to finish the user story incrementally step by step.

Based on the experience gained in the development of this project, it can be concluded that the application of the technique is a better alternative for software development, achieving a higher quality in delivered software product and the programmer can perform less work solving errors that can be produced in the future.

Introducción

En el presente documento se documenta la aplicación de la técnica TDD en un proyecto de software.

Actualmente hay poca documentación acerca de la implementación de la técnica en un caso real donde se describa la experiencia adquirida al implementarla lo cual se pudo evidenciar al realizar investigación y búsqueda sobre el tema.

Por eso se propone como objetivo documentar la aplicación de la técnica de desarrollo de software Test Driven Development con elementos de Scrum a través del desarrollo de un producto de software llamado ProyectoPlus para el seguimiento de proyectos de grado en la universidad Antonio José Camacho para mejorar esta situación.

En el documento se encuentran el desarrollo del producto mediante las fases de Scrum con la técnica de desarrollo de software Test Driven Development, describiendo la aplicación de esta práctica en la fase de implementación para tener una culminación exitosa del producto de software.

Finalizando con un producto de software puesto en marcha y una documentación que puede servir a los desarrolladores de software para orientar su aplicación de esta técnica en sus proyectos.

1. Planteamiento del Problema

Actualmente falta documentación acerca de la implementación de la técnica TDD en el desarrollo de software en el marco de una metodología ágil. Para realizar este proyecto se investigó acerca de la ejecución de la técnica TDD en casos reales y se encontró información en diversos portales y la mayoría de resultados son experiencias descritas a través de posts o blogs, pero los artículos científicos o documentación de proyectos son muy limitados y falta el desarrollo herramientas como material de trabajo independiente para los estudiantes o apasionados en el desarrollo de software que quieren profundizar en nuevas metodologías, lo cual servirá para involucrarse en el desarrollo mediante metodologías de desarrollo ágiles y eficientes.

De igual manera en la universidad Antonio Jose Camacho no se producen artículos orientados a la técnica de programación TDD, ya que es común denominador de los estudiantes de ingeniería generar estudios o proyectos bajo metodologías RUP.

Sin embargo la técnica TDD no es suficiente para generar un cambio significativo en un proyecto de software, es necesario combinar la técnica con el marco de una metodología ágil para producir un producto de software que sirva a los desarrolladores para orientar sus procesos de implementación con metodologías ágiles, ya que la metodología tradicional en fases está quedando obsoleta, (Quingaluisa y Gómez, 2015) afirma:

Con la incorporación del software en la gran mayoría de procesos organizacionales y la necesidad de obtener productos terminados en un menor tiempo, los requerimientos de desarrollo comenzaron a cambiar a un ritmo acelerado, volviendo obsoleta la metodología de desarrollo por fases que había imperado.

Adicionalmente, Autores como Girón Cuenca y Veintimilla, habla sobre la disminución de costos en producción y mantenibilidad del código manteniendo la calidad, citando lo siguiente:

Es cierto que la diversificación en el desarrollo de software se ha expandido, surgen nuevas prácticas, se requiere disminuir costos en producción y mantenibilidad del código aun manteniendo la calidad, esto genera las metodologías ágiles como una posible respuesta a los problemas del desarrollo de software. (Girón Cuenca y Veintimilla, s.f)

Apuntando a una disminución de costo y mantenimiento es ideal llegar a este punto desde la implementación del código a través de tests ya que esto permitirá disminuir los tiempos de desarrollo optimizando el valioso tiempo, contando como otro recurso importante en el desarrollo de proyectos de software a pesar de no ser algo tangible. (Vaca, y otros, 2013)

1.1 Pregunta de Formulación

¿Cómo generar material de apoyo para los desarrolladores de software que quieran implementar la técnica TDD combinada con una metodología ágil mediante el desarrollo de un software de proyectos de grados?

1.2 Pregunta de Sistematización

- ¿Cómo combinar la técnica Test Driven Development con una metodología ágil?
- ¿Cuáles son las ventajas de aplicar una práctica de Desarrollo guiado por pruebas frente al ciclo tradicional de testing?
- ¿Cómo demostrar las ventajas de la aplicación de TDD en el marco de una metodología?

2. Objetivos

2.1 Objetivo General

Documentar la aplicación de la técnica Test Driven Development con elementos de SCRUM mediante el desarrollo del software de seguimiento de proyectos de grados en la facultad de ingenierías de la UNIAJC

2.2 Objetivos Específicos

- Proponer un proceso de desarrollo aplicando Test Driven Development con las estrategias ágiles de SCRUM.
- Comparar las ventajas y desventajas de trabajar o no con la técnica de desarrollo guiado por pruebas a través del desarrollo de un sistema de gestión de proyectos de grados.
- Implementar la técnica propuesta de desarrollo Test Driven Development.

3. Justificación

A continuación se describe la justificación para realizar el proyecto ya que:

- Una de las necesidades más apremiantes es la de proporcionar a los programadores de software herramientas que les permita desarrollar su potencial a la hora de empezar a generar productos de software que cumpla con estándares de calidad, mantenibilidad, que estén abiertos a los cambios y que permita optimizar costos, todo esto con el fin de que se puedan desenvolver con éxito en la implementación de una aplicación, sea web, móvil o de escritorio.
- Los desarrolladores tendrán más tiempo para invertir en otros aspectos ya que la idea es reducir la pérdida de tiempo en tener que soportar un producto con errores, de esta forma se podrán enfocar en realizar más entregas que den valor y así poder tener un cliente satisfecho. A si mismo los clientes se verán beneficiados con productos con menos errores, fáciles de mantener y que se adaptan a las necesidades cambiantes de su negocio.
- Proporcionar a la comunidad de desarrolladores, profesores, estudiantes o personas que les interese un caso documentado del desarrollo de un software de control de proyectos de grado, aplicando la técnica Test Driven Development con elementos de una metodología ágil. Con esto, se tendrá una base documental con la cual se podrán apoyar para aplicar a sus futuros proyectos de implementación de software tanto para el ambiente educativo como también para el laboral.

4. Alcances y limitaciones

El presente proyecto explorará y propondrá una aplicación de una técnica de desarrollo denominada Test Driven Development (TDD) combinada con elementos de la metodología ágil SCRUM y tendrá como finalidad el estudio y la posterior documentación de todos los pasos y procesos que deben ser aplicados sobre la metodología de desarrollo para cumplir los requisitos solicitados por la universidad Antonio José Camacho y sus fines académicos.

4.1 Limitaciones

- El software producido en este proyecto, será propiedad de la universidad, los cuales lo definen para uso exclusivo de la misma.
- El software empleado para llevar a cabo la implementación, deberá ser de licencia libre, por lo cual los requerimientos del usuario y la metodología de desarrollo escogida deberán adaptarse para cumplir con esta limitante.

5. Marco Referencial

5.1 Antecedentes

(Kanewala & Bieman, 2014)

Testing scientific software: A systematic literature review

En el artículo publicado en mayo del 2014 se realiza un estudio en una gran cantidad de artículos científicos relacionados con el desarrollo de software científico, enfocándose principalmente en las pruebas a los que fueron sometidos los respectivos desarrollos realizados, analizando toda la información obtenida y enfocándola a una conclusión en cuanto a como se ha puesto en marcha todo el tema de la implementación de pruebas.

Se encuentra una problemática transversal entre programadores de software y científicos, donde se cruza la línea entre dos o más campos de las ciencias, donde surge siempre la gran inquietud en la cual se basa el artículo y es como puede ser sometido a pruebas un software científico, ya que los programadores de software discrepan con el enfoque que le quiere dar el científico al desarrollo del software surgiendo problemas al trabajar juntos, algunos de estos software científicos tienen como propósito simular el efecto del uso de armas sobre el ambiente, entre otros campos como el militar, software de simulación de salud, software desarrollado por biólogos y científicos espaciales, software para modelado cardiaco en biología computacional, los modelos climáticos, entre otros.

El eje central del artículo abarca el tema de cómo se han sometido a pruebas el software científico ayudando de esta forma a identificar los retos específicos que se enfrentan al probar el software donde se destaca un caso de estudio para el desarrollo de software para los modelos climáticos bajo el marco de la técnica TDD aunque también se encuentra que el software científico es desarrollado por los mismos científicos y por esto hace fácil aplicar determinadas técnicas para equipos multidisciplinarios. Cabe mencionar que algunos proyectos han utilizado Test Driven Development (TDD), donde se han escrito pruebas para verificar la funcionalidad antes de que el código se aplique.

Un antecedente que agrega un componente extra y clave para el desarrollo del actual proyecto en cuanto a la aplicación de pruebas que nos permite identificar los retos más comunes a la hora de generar casos de pruebas.

(Beck, Test-Driven Development By Example, 2002)

Test-Driven Development By Example

Se discute sobre los patrones de diseño o refactorización aplicados en la práctica de TDD, esto podría abarcar un libro completo para hablar de los patrones existentes como las pruebas aisladas, escribir una lista de pruebas, datos de prueba que se encuentran asociadas a la técnica de simulación de datos con el framework Mockito etc.

Al mismo tiempo se hace un recorrido sobre los patrones aplicados a TDD, pero el autor no entra a una gran profundidad en los patrones existentes. También es cierto que se debe saber cuándo se deben escribir las pruebas, donde se deben escribir las pruebas y cuando se debe dejar de escribir estas pruebas, haciendo parte esto también al término de Red Bar Patterns (patrones de barra roja) y después se pasa a los Green Bar Patterns (patrones de barra verde)

Aquí vale la pena resaltar que a la hora de escribir una prueba es importante realizarse una serie de preguntas como:

- ¿Dónde pertenece la operación?
- ¿Cuáles son las entradas correctas?
- ¿Cuál es la salida correcta dadas esas entradas?

Para el desarrollo de este proyecto este antecedente brinda estándares que se deben aplicar a la implementación del código del producto enfocado en la práctica de desarrollo guiado por pruebas TDD. También nos brinda los pasos a seguir en la creación de los determinados casos de prueba y el tratamiento que se le debe dar a cada prueba implícita en el ciclo que ofrece la técnica.

THE ART OF SOFTWARE TESTING

El Agilísimo busca tomar medidas para minimizar errores por lo tanto se habla de reuniones constantemente, donde deben ir usuarios y analistas del negocio. Las reuniones deben ser breves y por lo tanto deben tener una hora de inicio y una hora final establecidas antes de dar inicio a la reunión, se busca que todo sea muy claro y concreto para llegar a una decisión unánime.

Es importante que las decisiones de arquitectura las tome todo el equipo y no deben ser impuestas por el arquitecto de software. Se menciona que siempre es importante implementar patrones de diseño y el uso de las TDD (Desarrollo Dirigido por Test) ya que sirven de herramienta para el diseño del código y por lo tanto se deben escribir antes de dar inicio al desarrollo.

Este antecedente será importante para enfocar el desarrollo de los casos de prueba en un camino ágil y sobre todo que estas pruebas ayuden en el diseño del código del proyecto.

(Zbucki & Pons, 2015)

Una Propuesta de Implementación MDD y TDD en el Dominio de Sistemas de Procesamiento Transaccional.

Es una propuesta de implementación de MDD (Desarrollo dirigido por modelos) y TDD (Desarrollo dirigido por pruebas) en el Dominio de Sistemas de Procesamiento Transaccional.

La propuesta define una base de conceptos comunes, que han sido obtenidos de la experiencia de algunos desarrolladores y de unas soluciones preexistentes.

Este proyecto busca que en conjunto de esas dos prácticas de desarrollo de software se establezca un Framework orientado al dominio que fomente la reutilización de código por medio de la refactorización y la implementación de las pruebas unitarias, implementando la práctica TDD.

Este antecedente, reduce la brecha de desconocimiento en la implementación de TDD junto con otras metodologías, mostrando otras experiencias importantes en la ejecución de proyectos transaccionales.

(Godoya, y otros, 2012)

Simulando Proyectos de Desarrollo de Software que incorporan Test Driven Development

En este artículo se presenta una línea de investigación denominada Simulación como Herramienta para la Mejora de los Procesos de Software Desarrollados con Metodologías Ágiles Utilizando Dinámica de Sistemas, cuyo objetivo es construir modelos de simulación utilizando la disciplina de dinámica de sistemas, que permitan estudiar los procesos de desarrollo de software ágiles gestionados por diversas metodologías y prácticas ágiles.

Estos antecedentes brindan un Modelo de Simulación Dinámico para estudiar la incorporación de prácticas de Test Driven Development a los proyectos de software. Las encuestas realizadas en este artículo, brindan información importante sobre la acogida de la implementación de la técnica, TDD en una comunidad de desarrolladores de software.

5.2 Marco Conceptual

5.2.1 Test Driven Development (TDD).

Una práctica de desarrollo de software que se enfoca en Diseñar las pruebas de software antes de iniciar con el proceso de crear el código, el cual debe superar las pruebas diseñadas para él. (Beck, Test-Driven Development By Example, 2002)

5.2.2 Acceptance Test Driven Development (ATDD)

Nivel de TDD en el cual todo el equipo discute en colaboración criterios de aceptación, con ejemplos, y luego los divide en un conjunto de pruebas de aceptación en concreto antes de que comience el desarrollo. (Jurado, 2010)

5.2.3 Developer TDD

Nivel donde se diseñan las pruebas unitarias. Su objetivo es especificar a detalle, el diseño ejecutable para la solución. (Jurado, 2010)

5.3 Marco Teórico

La técnica TDD (Test Driven Development) se compone de la siguiente estructura.

Figura 1 Ciclo de vida TDD



5.3.1 Fases TDD

La práctica de TDD se divide en 3 fases, como se puede apreciar en la Figura 1 Ciclo de vida TDD. (Jurado, 2010)

- **Escribir las pruebas:** Esta etapa del ciclo también se le conoce como Rojo, hacer que la prueba falle. Las pruebas se escriben antes de empezar a escribir el código, la prueba debe buscar que haya una falla, estas son escritas por los propios desarrolladores una vez tengan claro el requisito y se busca que los mismos logren un entendimiento de lo que deben desarrollar mediante la construcción del código con el que se va a probar el test pase.

Se deben escribir en forma de Test unitarios [14] ya que se probara el código que satisface el test, usualmente se utiliza un framework xUnit, pero xUnit no es una herramienta en sí misma. La letra x es tan solo un prefijo a modo de comodín, ya que son numerosos frameworks basados en el original SUnit. Sunit fue creado por Kent Beck para la plataforma SmallTalk y se ha llevado a otros lenguajes como Java, .Net, Python, Ruby, Perl, C++, etc.

- **Implementar el código que hace funcionar el ejemplo:** Esta etapa del ciclo también se le conoce como Verde, porque las pruebas no deben fallar al momento de implementar el código.

Una vez que se escribe el test unitario en el framework elegido de acuerdo al lenguaje sobre el que se implementara la aplicación, se procede a escribir el código estrictamente necesario para que el test no genere error, y el cual hace énfasis en la expresión “Keep It Simple, Stupid!” (Déjelo Simple), se debe ignorar si el código se ve feo.

Si bien se vendrán dudas al momento de implementar el código referente al comportamiento concerniente a ciertas variables de entrada y salida, esto se debe escribir para tenerlo en cuenta en una próxima iteración, más no se debe desarrollar en la iteración actual.

- **Refactorización del código:** Permite mantener la calidad de la arquitectura, se cambia el diseño sin cambiar la funcionalidad, y no debe afectar la ejecución de los tests unitarios. Se buscan las líneas duplicadas y se eliminan, aplicando la refactorización.

5.3.2 ATDD (Acceptance Test Driven Development)

Con respecto a ATDD, se pueden definir las siguientes características (Jurado, 2010):

- **Historias de Usuario.** Se escribe el requisito especificando el rol, lo que requiero y para que lo requiero, luego se escriben los ejemplos que vendrían siendo los criterios de aceptación, esto se lleva a cabo por medio del artefacto Historias de Usuario y Criterios de aceptación.
- **Discutir.** Un probador, desarrollador y experto en dominios se sientan con el propietario del producto y discuten la historia, descomponiéndola y generando las especificaciones o criterios de aceptación.
- **Decidir.** El propietario del producto decide que las especificaciones cubren el comportamiento.
- **Desarrollo.** El equipo de tester anota las especificaciones olvidadas para la siguiente iteración, donde se podrán priorizar apropiadamente.
- **El equipo de developers,** desarrollan las especificaciones instrumentadas, crean pruebas de fallos y luego implementan la funcionalidad, utilizando el artefacto de historias de usuario con sus especificaciones o criterios de aceptación y también el fixture usando Frameworks como el Fitnesse y Cucumber para la automatización de las pruebas.
- **Demostrar.** Una vez que todas las pruebas pasan, la historia puede ser demostrada al dueño del producto.

5.3.3 TDD (Test Driven Development) paso a paso

A continuación se describe la aplicación de la técnica TDD en varios pasos. (Beck, Test-Driven Development By Example, 2002)

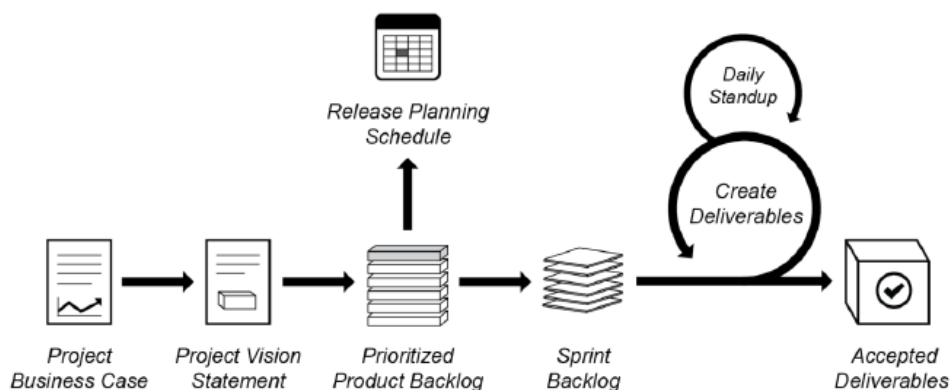
- **Write a test (Escriba una Prueba):** En este paso se escribe el requisito en forma de Test. Se pueden utilizar frameworks de testing como JUnit para Java, NUnit para .NET, PYunit para Python, PHPUnit para PHP entre otros.
- Se deben escribir los Test Unitarios por cada método que se ha realizado para los criterios de aceptación.
- **Test de Aceptación:** donde se asegura que se ha desarrollado la funcionalidad adecuada.
- **Run tests see new failure (Verificar si la prueba falla):** Verificar si la prueba falla, esto garantiza la calidad de la implementación, ya que en el caso de que la prueba falla, se volverá a escribir la prueba, volviendo al primer paso.
- **Write some code (Escribir el código):** Una vez que se tienen los test, se procede a escribir el código estrictamente necesario para que se cumpla el requerimiento donde se hace referencia a la expresión “Keep It Simple, Stupid!” (Déjelo Simple), se debe ignorar si el código se ve feo. Si bien se vendrán dudas al momento de implementar el código referente al comportamiento referente a ciertas variables de entrada y salida, esto se debe escribir para tenerlo en cuenta en una próxima iteración, más no se debe desarrollar en el momento.
- **Run tests see all pass (Ejecutar la prueba y verificar que funciona):** Verificar si la prueba funciona correctamente con el código escrito en el paso anterior. Se puede usar herramientas como el Selenium ya que es un entorno de pruebas para aplicaciones basadas en web que facilitara la ejecución del código implementado.
- **Refactor (Refactorización):** En este paso se realizará la refactorización del código, lo cual será la modificación del diseño sin alterar su comportamiento. Se buscan las líneas duplicadas y se eliminan, aplicando la refactorización.
- Es importante revisar patrones de diseño, una técnica útil a implementar es S.O.L.I.D que son cinco principios básicos de la programación orientada a objetos y el diseño. Cuando estos principios se aplican en conjunto es más probable que un desarrollador cree un sistema que sea fácil de mantener y ampliar con el tiempo.

5.3.4 SCRUM

Es una metodología que sirve para gestionar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Los ciclos entregan piezas de software que le agregan valor al proyecto. Las iteraciones en general tienen una duración entre 2 y 4 semanas, en grupos de 3 a 9 personas.

Por una parte, la metodología SCRUM inicia con un Project Business Case, donde se crea una visión general del proyecto, a continuación se crea un Backlog del producto priorizado lo cual contiene una lista priorizada de los requerimientos del negocio en forma de Historias de Usuarios. El inicio de un sprint comienza con el Sprint Planning Meeting donde las historias de usuarios consideradas como alta prioridad se incluyen en el Sprint. Durante el Sprint se trabaja en los entregables y se realizan reuniones diarias para que el equipo conozca cómo va la el desarrollo del Sprint. A medida que se concluye cada Sprint, el dueño del Product Owner y los Stakeholders revisan el respectivo incremento y el Product Owner acepta si estos cumplen con los criterios de aceptación. El ciclo del Sprint termina con un Retrospect Sprint Meeting, en efecto se puede apreciar el flujo de SCRUM en la siguiente figura.

Figura 2 Flujo de SCRUM



(2013). SCRUMstudy. [Figura].

5.3.4.1 Fases de SCRUM

INCIAR

En la fase de inicio, se crea la visión del proyecto, se crea el equipo Scrum, se generan las épicas y se crea el Product Backlog Priorizado

PLANEAR Y ESTIMAR

En la fase de planear y estimar se crean las historias de usuarios y se priorizan, estiman y aprueban las historias de usuarios y se crea el Sprint Backlog con sus respectivas tareas a realizar.

IMPLEMENTAR

En esta fase se crean los entregables que son de mayor valor para el producto y se realizan reuniones diarias para evaluar los avances e impedimentos adquiridos durante el Sprint, aquí el equipo Scrum responde preguntas como. ¿Qué hice ayer?, ¿qué estoy haciendo hoy? y ¿qué impedimentos tengo para realizar mi trabajo?

REVISIÓN Y RETROSPECTIVA

En esta fase se demuestran y se validan los Sprints con el Product Owner y se lleva a cabo una reunión con el equipo Scrum para hacer las retrospectivas del Sprint, se discute lo que se hizo, las buenas prácticas adquiridas y lo que no se debe hacer en un próximo Sprint.

LANZAMIENTO

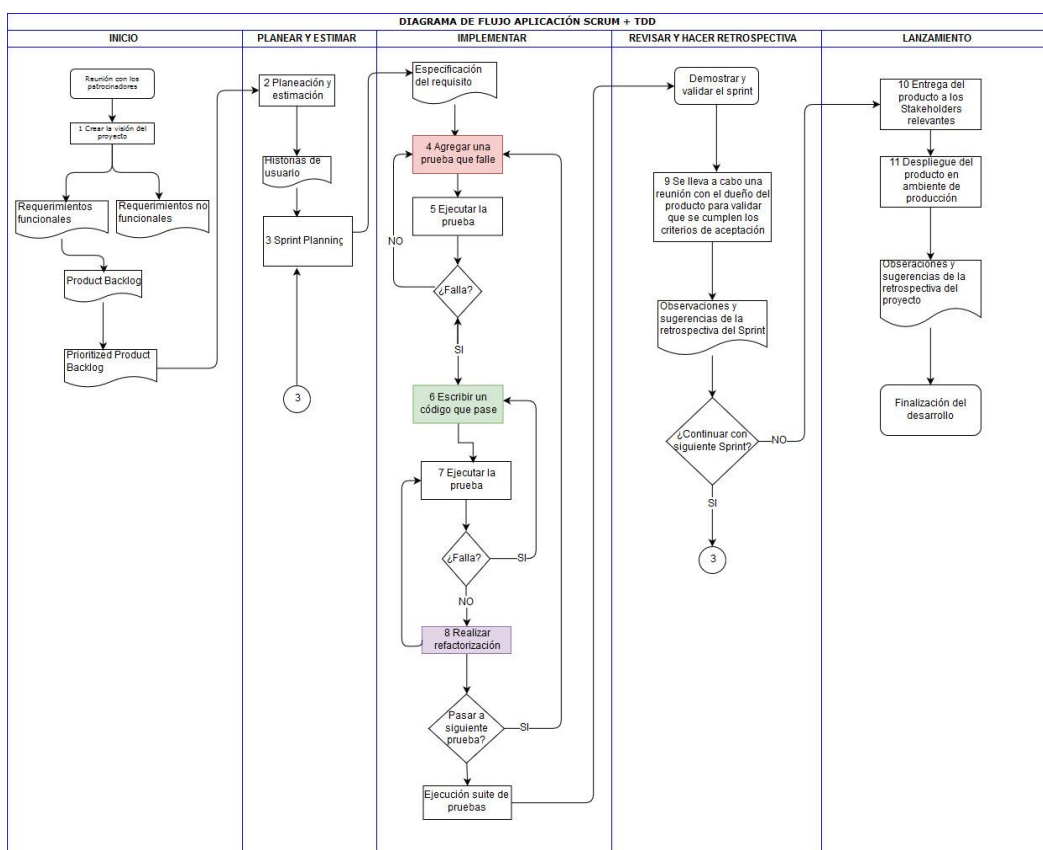
La fase de lanzamiento se centra en la entrega de los entregables aceptados al cliente y en la identificación, documentación e internalización de las lecciones aprendidas durante el proyecto. (SCRUMstudy, 2013)

6. Desarrollo

Se llevará a cabo el desarrollo completo de la aplicación ProyectoPlus para la universidad Antonio José Camacho implementando artefactos de la metodología Scrum, por medio de las fases de Inicio, Planeación y Estimación, Implementación, Revisión y Retrospectiva y finalmente Lanzamiento.

El ciclo de vida TDD que se va a aplicar en este proyecto se encuentra dentro de la fase de IMPLEMENTAR del marco de la metodología SCRUM. A continuación se detalla el desarrollo de la aplicación en el siguiente diagrama.

Figura 3 Diagrama de SCRUM con TDD



6.1 Inicio

6.1.1 Crear la visión del proyecto.

En la primera reunión se establece la visión del proyecto y se realizan unos requerimientos funcionales en los cuales se enmarcan las funcionalidades de la aplicación, la aplicación debe facilitar la gestión de los proyectos de grado para los estudiantes de últimos semestres de la universidad Antonio José Camacho, poder llevar un seguimiento desde que se genera la ficha de anteproyecto hasta que el mismo es aprobado y evaluado por el jurado para ser un proyecto de grado.

6.1.2 Requerimientos Funcionales

Tabla 1 *Requerimientos Funcionales*

Código	Descripción	Prioridad
RF01	El sistema debe permitir registrar evaluaciones de sustentación de proyecto de grado	ALTO
RF02	El sistema debe permitir calcular la nota final al ingresar las notas de los estudiantes.	ALTO
RF03	El sistema debe permitir gestionar los administradores del sistema.	ALTO
RF04	El sistema debe permitir gestionar los estudiantes.	ALTO
RF05	El sistema debe permitir gestionar los docentes.	ALTO
RF06	El sistema debe tener un login con usuario y contraseña que permitirá autenticar a los usuarios.	ALTO

RF07	El sistema debe contener una página principal en la cual exista un menú y/o enlaces que permitan acceder a las diferentes rutas.	ALTO
RF08	El sistema debe permitir crear fichas de anteproyecto de grado	ALTO
RF09	El sistema debe permitir consultar las fichas de anteproyecto de grado.	ALTO
RF10	El sistema debe permitir crear actas de cambios sobre una ficha de anteproyecto ya cerrada.	ALTO
RF11	El sistema debe permitir llevar un registro de cambios de objetivos específicos, anotando unas observaciones así la ficha de anteproyecto haya sido cerrada.	ALTO
RF12	El sistema debe permitir generar el reporte de la ficha de anteproyecto de grado, por temática, evaluadores, programa, director del proyecto.	ALTO
RF13	El sistema debe permitir adjuntar el proyecto de grado en formato PDF	BAJO
RF14	El sistema debe permitir asignar un director del proyecto por el administrador.	BAJO
RF15	El sistema debe permitir tiempo de corrección si el anteproyecto es rechazado	BAJO
RF16	Enviar correo al director del proyecto una vez sea asignado al proyecto.	BAJO
RF17	El sistema debe permitir subir el cronograma en PDF.	BAJO

RF18	El sistema debe permitir ingresar hasta 3 evaluadores para el proyecto.	MEDIO
RF19	El sistema debe permitir registrar el <u>acta de sustentación</u> de proyecto de grado	MEDIO
RF20	El sistema debe permitir cerrar fichas de anteproyectos que estén evaluados.	MEDIO
RF21	El sistema debe permitir actualizar las fichas de anteproyectos hasta el tiempo de entrega.	ALTO
RF22	El sistema debe permitir evaluar la ficha de anteproyecto de grado y debe quedar de solo consulta una vez se evalué.	ALTO
RF23	El sistema debe permitir la revisión de anteproyecto de grado.	ALTO
RF24	El sistema debe permitir eliminar una ficha de anteproyecto que este en estado pendiente.	BAJO

6.1.3 Requerimientos No Funcionales

Tabla 2 *Requerimientos No Funcionales*

Código	Descripción	Prioridad
RNF01	El portal debe ser desarrollado mediante el Framework de Laravel	ALTA
RNF02	El Portal Web debe tener un diseño Responsive, para que se pueda adaptar a resoluciones de diferentes dimensiones.	ALTA
RNF03	El portal web debe ser alojado en un hosting de mínimo 1 GB.	BAJA
RNF04	El servidor debe contar con PHP 5.6	BAJA

RNF05	El servidor debe contar con una base de datos Postgresql 9,6 o una versión posterior.	BAJA
-------	---	------

6.1.4 Product Backlog

En esta parte del proyecto, se genera el product backlog inicial.

Tabla 3 *Product Backlog*

Identificador (ID) de la Historia	Enunciado de la Historia
HU-001	Yo, como estudiante, necesito crear la ficha de anteproyecto, con la finalidad de que esta pueda ser revisada por los evaluadores.
HU-002	Yo, como jurado, necesito evaluar el proyecto de los estudiantes, con la finalidad de que el proyecto sea evaluado y pueda finalizarse.
HU-003	Yo, como administrador, necesito crear los usuarios del sistema, con la finalidad de guardar nuevos usuarios, como administradores, estudiantes y docentes.
HU-004	Yo, como administrador, necesito editar los usuarios del sistema, con la finalidad de cambiar el rol y la información de los usuarios.
HU-005	Yo, como administrador, necesito eliminar los usuarios del sistema, con la finalidad de que estos sean dados de baja.
HU-006	Yo, como administrador, necesito listar los usuarios del sistema, con la finalidad encontrar los usuarios registrados en el sistema y poder gestionarlos
HU-007	Yo, como usuario, necesito poder autenticarme en el sistema, con la finalidad de poder navegar en todas las opciones disponibles para mi rol.

HU-008	Yo, como docente necesito evaluar el proyecto de los estudiantes, con la finalidad de que la ficha quede revisada y con observaciones para que pueda ser evaluada por jurados.
HU-009	Yo, como director del programa, necesito listar, editar y/o anular las fichas de anteproyecto, con la finalidad de ver las fichas de anteproyectos creadas en el sistema, editarlas y/o anularlas.
HU-010	Yo, como estudiante, necesito, consultar las fichas de anteproyecto asociadas a los estudiantes autenticados en el sistema, con la finalidad de poder, conocer el detalle de la ficha de anteproyecto y poder editarla o eliminarla.
HU-011	Yo, como director del programa, necesito generar un reporte de la ficha, con la finalidad de, conocer la información de las fichas y sus estados y hacer análisis de la información.
HU-012	Yo, como docente, necesito recibir un correo de notificación, con la finalidad de saber de qué fui asignado como director de un proyecto.
HU-013	Yo, como director del programa, necesito cerrar la ficha de anteproyecto, con la finalidad de que la ficha no se pueda editar.
HU-014	Yo, como estudiante, necesito editar la ficha de anteproyecto, con la finalidad de poder corregir y/o completar la información de la ficha.
HU-015	Yo, como director de proyecto, necesito editar la ficha de anteproyecto, con la finalidad de poder asignar los docentes evaluadores.
HU-016	Yo, como estudiante, necesito eliminar una ficha creada, con la finalidad de poder dar de baja a una ficha en el sistema
HU-017	Yo, como director del programa, necesito crear una nueva evaluación del proyecto, con la finalidad de que se pueda realizar la evaluación cuantitativa del proyecto de grado.
HU-018	Yo, como director de proyecto o docente, necesito listar, editar las fichas de anteproyecto o editar la evaluación, con la finalidad de ver las fichas de anteproyectos creadas en el sistema, editarlas y/o editar la evaluación

HU-019	Yo, como analista de desarrollo, necesito listar montar un arquetipo, con la finalidad de que la tecnología a implementar se integre y poder empezar a desarrollar.
--------	---

6.1.5 Priorización del Product Backlog.

Se realiza un refinamiento y priorización de los requisitos del Product Backlog sobre la base del valor del producto.

Se emplea la técnica MosCow Prioritization Scheme, el cual deriva su nombre de las primeras letras de las frases “must have” (debe tener), “should have” (debería tener), “could have” (podría tener) y “will not have” (no tendrá)

Las etiquetas están en orden decreciente de prioridad, donde “must have” (debe tener) son aquellos sin los que el producto no tendrá valor y “will not have” (no tendrá) son aquellos que, a pesar de que sería bueno tener, no son necesarios para que sean incluidos.

6.1.6 Prioritized Product Backlog

Tabla 4 *Prioritized Product Backlog*

Identificador (ID) de la Historia	Enunciado de la Historia	Prioridad
HU-001	Yo, como estudiante, necesito crear la ficha de anteproyecto, con la finalidad de que esta pueda ser revisada por los evaluadores.	MUST
HU-003	Yo, como administrador, necesito crear los usuarios del sistema, con la finalidad de guardar	MUST

	nuevos usuarios, como administradores, estudiantes y docentes.	
HU-004	Yo, como administrador, necesito editar los usuarios del sistema, con la finalidad de cambiar el rol y la información de los usuarios.	MUST
HU-005	Yo, como administrador, necesito eliminar los usuarios del sistema, con la finalidad de que estos sean dados de baja.	MUST
HU-006	Yo, como administrador, necesito listar los usuarios del sistema, con la finalidad encontrar los usuarios registrados en el sistema y poder gestionarlos	MUST
HU-007	Yo, como usuario, necesito poder autenticarme en el sistema, con la finalidad de poder navegar en todas las opciones disponibles para mi rol.	MUST
HU-008	Yo, como docente necesito evaluar el proyecto de los estudiantes, con la finalidad de que la ficha quede revisada y con observaciones para que pueda ser evaluada por jurados.	MUST
HU-010	Yo, como estudiante, necesito, consultar las fichas de anteproyecto asociadas a los estudiantes autenticados en el sistema, con la finalidad de poder, conocer el detalle de la ficha de anteproyecto y poder editarla o eliminarla.	MUST
HU-014	Yo, como estudiante, necesito editar la ficha de anteproyecto, con la finalidad de poder corregir y/o completar la información de la ficha.	MUST
HU-015	Yo, como director de proyecto, necesito editar la ficha de anteproyecto, con la finalidad de poder asignar los docentes evaluadores.	MUST

HU-016	Yo, como estudiante, necesito eliminar una ficha creada, con la finalidad de poder dar de baja a una ficha en el sistema	MUST
HU-017	Yo, como director del programa, necesito crear una nueva evaluación del proyecto, con la finalidad de que se pueda realizar la evaluación cuantitativa del proyecto de grado.	MUST
HU-018	Yo, como director de proyecto o docente, necesito listar, editar las fichas de anteproyecto o editar la evaluación, con la finalidad de ver las fichas de anteproyectos creadas en el sistema, editarlas y/o editar la evaluación	MUST
HU-019	Yo, como analista de desarrollo, necesito listar montar un arquetipo, con la finalidad de que la tecnología a implementar se integre y poder empezar a desarrollar.	MUST
HU-002	Yo, como jurado, necesito evaluar el proyecto de los estudiantes, con la finalidad de que el proyecto sea evaluado y pueda finalizarse.	SHOULD
HU-009	Yo, como director del programa, necesito listar, editar y/o anular las fichas de anteproyecto, con la finalidad de ver las fichas de anteproyectos creadas en el sistema, editarlas y/o anularlas.	SHOULD
HU-013	Yo, como director del programa, necesito cerrar la ficha de anteproyecto, con la finalidad de que la ficha no se pueda editar.	SHOULD
HU-011	Yo, como director del programa, necesito generar un reporte de la ficha, con la finalidad de, conocer la información de las fichas y sus estados y hacer análisis de la información.	COULD

HU-012	Yo, como docente, necesito recibir un correo de notificación, con la finalidad de saber de qué fui asignado como director de un proyecto.	COULD
--------	---	-------

6.2 Planear y Estimar

6.2.1 Historias de usuario

En este proceso se crean las historias de usuario y los criterios de aceptación con el fin de asegurar que los requisitos se encuentran claramente representados y que puedan ser plenamente comprendidos.

Figura 4 Historias de Usuario

ID HU	Rol	ENUNCIADO			CRITERIOS DE ACEPTACIÓN		
		Descripción de la Funcionalidad	Descripción de la Consecuencia	Número de Escenario	Contexto	Evento	Resultado
HU-001	Yo, como estudiante	Necesito crear la ficha de anteproyecto	Con la finalidad de que esta pueda ser revisada por los evaluadores.	1	En caso de que el estudiante vaya a empezar con su anteproyecto de grado.	Cuando el estudiante guarde la ficha.	El sistema desplegará un formulario donde se ingresan los datos de la ficha de anteproyecto, esa información será guardada en la base de datos y redirecciona al listado de usuarios.
				2	En caso de que el estudiante no se haya autenticado en el sistema.	El usuario ingresa por la url para crear una ficha	El sistema redirecciona al login, para que la persona que intenta acceder, se pueda autenticar en el sistema.
				3	En caso de que se adjunte un pdf	El estudiante selecciona un pdf con el cronograma.	El sistema guarda el documento pdf en el servidor.
				4	En caso de que el usuario envíe el formulario.	Cuando el estudiante guarde la ficha.	El sistema solicita que se llenen los campos obligatorios.
				5	En caso de que se adjunte un pdf	El estudiante selecciona un pdf con el proyecto.	El sistema guarda el documento pdf en el servidor.
				6	El estudiante que ingrese al formulario debe tener el rol de estudiante	El usuario ingresa por la url para crear una ficha	El sistema redirecciona al login, para que la persona que intenta acceder, se pueda autenticar en el sistema.
HU-002	Yo, como jurado	Necesito evaluar el proyecto de los estudiantes.	Con la finalidad de que el proyecto sea evaluado y pueda finalizarse.	1	La nota final es calculada automáticamente por el sistema.	Cuando el jurado ingrese todas las notas de los criterios establecidos	El sistema calcula la nota final y la coloca en pantalla para que sea guardada junto con los demás datos.
				2	Nota igual o superior a 3,0	Cuando el jurado ingrese todas las notas de los criterios establecidos	El sistema guarda la evaluación y cambia el estado a APROBADO
				3	La nota es igual o superior a 2,0 y menor que 3,0	Cuando el jurado ingrese todas las notas de los criterios establecidos	El sistema guarda la evaluación y cambia el estado a PENDIENTE
				4	La nota es inferior a 2,0	Cuando el jurado ingrese todas las notas de los criterios establecidos	El sistema guarda la evaluación y cambia el estado a RECHAZADO
HU-003	Yo, como administrador	Necesito crear los usuarios del sistema	Con la finalidad de guardar nuevos usuarios, como administradores, estudiantes y docentes.	1	En caso de que quiera crear un nuevo usuario en el sistema	Cuando selecciona la opción de crear un usuario	El sistema despliega el formulario de creación del usuario, adicionalmente permite asociarle un rol que puede ser estudiante, docente o director de programa, luego guarda en la base de datos.
				2	En caso de que quiera crear un nuevo usuario en el sistema	Cuando selecciona la opción de crear un usuario	El sistema valida que no exista un usuario repetido
				3	En caso de que quiera crear un nuevo usuario en el sistema	Cuando selecciona la opción de crear un usuario	El sistema encripta la contraseña con un Hash

HU-004	Yo, como administrador	Necesito editar los usuarios del sistema	Con la finalidad de cambiar el rol y la información de los usuarios.	1	En caso de que quiera editar un usuario en el sistema	Cuando selecciona la opción de editar un usuario	El sistema valida que no exista un usuario repetido
				2	En caso de que quiera editar un usuario en el sistema	Cuando selecciona la opción de editar un usuario	el sistema despliega un formulario con los datos del usuario, se pueden cambiar la información y posteriormente se va a guardar la nueva información en la base de datos.
				3	En caso de que quiera editar un usuario en el sistema	Cuando selecciona la opción de editar un usuario	El sistema encripta la contraseña con un Hash
HU-005	Yo, como administrador	Necesito eliminar los usuarios del sistema	Con la finalidad de que estos sean dados de baja.	1	El usuario no se encuentra asociado a otras entidades	El administrador elimina el registro	El sistema borra de la base de datos el registro.
				2	El usuario se encuentra asociado a un estudiante o docente o director de programa.	El administrador elimina el registro	El sistema arroja un mensaje de error por que el usuario se encuentra asociado a otras entidades como estudiantes, docentes o directores.
HU-006	Yo, como administrador	Necesito listar los usuarios del sistema	Con la finalidad encontrar los usuarios registrados en el sistema y poder gestionarlos	1	Existen usuarios creados	El administrador ingresa a la sección de usuarios.	el sistema arroja un listado de usuarios creados en la base de datos
				2	No existen usuarios creados.	El administrador ingresa a la sección de usuarios.	El sistema muestra un mensaje, no se encontraron fichas asociadas.
HU-007	Yo, como usuario	Necesito poder autenticarme en el sistema.	Con la finalidad de poder navegar en todas las opciones disponibles para mi rol.	1	En caso de que sea un usuario administrador	cuando se autentique en el sistema.	el sistema direcciona al usuario al panel de administración
				2	En caso de que sea un usuario docente	cuando se autentique en el sistema.	el sistema direcciona al usuario al panel de docentes.
				3	En caso de que sea un estudiante	cuando se autentique en el sistema.	el sistema direcciona al usuario al panel de estudiantes
				4	En caso de que sea un director de programa	cuando se autentique en el sistema.	el sistema direcciona al usuario al panel de director de programa

HU-008	Yo, como director	Necesito evaluar el proyecto de los estudiantes	Con la finalidad de que la ficha quede revisada y con observaciones para que pueda ser evaluada posteriormente.	1	En caso que el docente va a evaluar la ficha	Cuando el docente evalúa desde el formulario de evaluación	El sistema solicita las observaciones y los evaluadores, después guarda la información y cambia el estado a corrección de ficha o revisado.
				2	En caso que se requiera revisar la ficha.	Cuando el docente visualiza el formulario para editar la ficha.	El sistema despliega los datos de la ficha en modo lectura excepto las observaciones y la asignación de evaluadores.
				3	En caso que se requiera revisar la ficha.	Cuando el docente visualiza el formulario para editar la ficha.	El sistema redirecciona al docente al listado de fichas una vez es revisado.
HU-009	Yo, como director del programa	Necesito listar, editar y/o anular las fichas de anteproyecto.	Con la finalidad de ver las fichas de anteproyectos creadas en el sistema, editarlas y/o anularlas.	1	Existen fichas de anteproyecto creadas en el sistema.	Cuando seleccione la opción de listar y editar las fichas de anteproyecto.	El sistema lista las fichas de anteproyecto con la opción de editar y eliminar cada ficha, si va a editar se despliega un formulario con los datos actuales a modo de lectura y se puede cambiar el director de proyecto. Al anular, cambia de estado en la base de datos.
				2	No existen fichas de anteproyecto en el sistema.	Cuando visita la sección de fichas asociadas.	El sistema muestra un mensaje, no se encontraron fichas.
HU-010	Yo, como estudiante	Necesito consultar las fichas de anteproyecto asociadas a los estudiantes autenticados en el sistema	Con la finalidad de poder conocer el detalle de la ficha de anteproyecto y poder gestionarlo.	1	En caso de que existan asignadas a los estudiantes autenticados en el sistema.	Cuando seleccione la opción de listar las fichas de anteproyecto.	El sistema lista las fichas de anteproyecto.
				2	No existen fichas de anteproyecto en el sistema.	Cuando seleccione la opción de listar las fichas de anteproyecto.	El sistema muestra un mensaje, no se encontraron fichas.
HU-011	Yo, como director del programa	Necesito generar un reporte de la ficha	Con la finalidad de conocer la información de las fichas y sus estados y hacer análisis de la información.	1	En caso de que el director del programa quiera consultar el reporte	Cuando selecciona la opción de ver el reporte de fichas.	El sistema muestra un reporte con la información detallada de las fichas de anteproyectos y evaluaciones respectivas
HU-012	Yo, como docente	Necesito recibir un correo de notificación	Con la finalidad de saber de que fui asignado como director de un proyecto.	1	En caso que el administrador modifique la ficha.	Cuando se guarde el formulario con el director asignado.	El sistema envía un correo de notificación al docente, primero valida que tenga un correo electrónico.
HU-013	Yo, como director del programa	Necesito cerrar la ficha de anteproyecto	Con la finalidad de que la ficha no se pueda editar.	1	En caso de que la ficha se encuentre en estado aprobado	Cuando seleccione la opción de consultar fichas asociadas	El sistema permite cambiar el estado a cerrado mientras este sea el director del programa.
				2	En caso de que no se encuentre en estado aprobado	Cuando seleccione la opción de consultar fichas asociadas	El sistema no permite cambiar el estado a cerrado.

HU-014	Yo, como estudiante	Necesito editar la ficha de anteproyecto	Con la finalidad de poder corregir y/o completar la información de la ficha.	1	En caso en que la ficha se encuentre en estado creado	Cuando el estudiante ingresa a la opción editar	El sistema despliega un formulario con los datos cargados, se guardan los datos en la base de datos.
				2	En caso de que el estudiante no se haya autenticado en el sistema.	El usuario ingresa por la url para editar una ficha	El sistema redirecciona al login, para que la persona que intenta acceder, se pueda autenticar en el sistema.
				3	En caso de que se adjunte un pdf	El estudiante selecciona un pdf con el cronograma.	El sistema guarda el documento pdf en el servidor.
				4	En caso de que el usuario envíe el formulario.	Cuando el estudiante guarde la ficha.	El sistema solicita que se llenen los campos obligatorios.
				5	En caso de que se adjunte un pdf	El estudiante selecciona un pdf con el proyecto.	El sistema guarda el documento pdf en el servidor.
				6	En caso de que el estudiante seleccione la pestaña Objetivos	El usuario desea agregar un objetivo específico	El sistema despliega un formulario para crear el objetivo específico y este se puede listar.
				7	En caso de que el estudiante seleccione la pestaña Objetivos	El usuario desea eliminar un objetivo específico	El sistema elimina de la base de datos el objetivo, mientras este aun no tenga observaciones del evaluador.
				8	En caso de que el estudiante seleccione la pestaña Objetivos	El usuario desea modificar un objetivo específico	El sistema despliega un formulario para editar el objetivo específico.
HU-015	Yo, como director de proyecto	Necesito editar la ficha de anteproyecto	Con la finalidad de poder asignar los docentes evaluadores.	1	En caso de que el administrador asigne los docentes	Cuando registra el docente	El sistema valida que solo se pueda asignar hasta 3 docentes que evalúan
				2	En caso que el administrador quiera editar una ficha	Cuando ingresa al formulario donde edita la ficha	El sistema desplegará un formulario pero los campos no podrán ser modificados, solo será posible asignar los docentes evaluadores.
HU-016	Yo, como estudiante	Necesito eliminar una ficha creada	Con la finalidad de poder dar de baja a una ficha en el sistema	1	En caso de que el estudiante tenga una ficha creada	Cuando el estudiante elimine la ficha	El sistema valida que la ficha se encuentre en estado pendiente y elimina el registro.
				2	En caso de que el estudiante tenga una ficha creada	Cuando el estudiante elimine la ficha	El sistema retorna un error por que la ficha esta en proceso de gestión y debe ser anulada por el administrador.

HU-017	Yo, como director del programa	Necesito crear una nueva evaluación del proyecto.	Con la finalidad de que se pueda realizar la evaluación cuantitativa del proyecto de grado.	1	En caso de que la ficha se encuentre en estado "a revisión"	Cuando el director del programa seleccione la opción de crear evaluación	El sistema despliega un formulario para asignar el jurado.
				2	En caso de que no se encuentre en estado "a revisión"	Cuando el director del programa visualice las fichas.	La ficha no tiene la opción para crear una nueva evaluación
HU-018	Yo, como director de proyecto ó docente	Necesito listar, editar las fichas de anteproyecto ó editar la evaluación.	Con la finalidad de ver las fichas de anteproyectos creadas en el sistema, editarlas y/o editar la evaluación	1	Existen fichas de anteproyecto creadas en el sistema.	Cuando seleccione la opción de consultar fichas asociadas	El sistema lista las fichas de anteproyecto asignadas al docente o director de proyecto que se encuentra autenticado en el sistema.
				2	No existen fichas de anteproyecto en el sistema.	Cuando seleccione la opción de consultar fichas asociadas	El sistema muestra un mensaje, no se encontraron fichas.
				3	En caso de que la ficha aun no tenga la evaluación creada	Cuando seleccione la opción de consultar fichas asociadas	El sistema no muestra la opción de editar evaluación.
				4	En caso de que la ficha se encuentre en estado aprobado ó cerrado	Cuando seleccione la opción de consultar fichas asociadas	El sistema permite generar el acta de sustentación.
HU-019	Yo, como analista de desarrollo	Necesito listar montar un arquetipo	Con la finalidad de que la tecnología a implementar se integre y poder empezar a desarrollar.	1	El ambiente de desarrollo debe funcionar con PHP 5.6.33 o superior	Cuando se ingrese a la URL http://localhost	El servidor responde la petición
				2	Instalación de Composer	NA	NA
				3	Instalar base de datos	NA	NA
				4	Instalar Bootstrap	NA	NA
				5	Instalar PHPUnit y Katalon para pruebas unitarias	NA	NA

6.2.2 Aprobar y estimar historias de usuarios.

Antes de realizar una planeación de todos los sprints, se realiza la respectiva aprobación y estimación de las historias de usuario.

Figura 5 Aprobación y Estimación de Historias de Usuario

Identificador (ID) de la Historia	Enunciado de la Historia	Estado	Estimación Poker	Iteración (Sprint)	Prioridad
HU-001	Yo, como estudiante, necesito crear la ficha de anteproyecto, con la finalidad de que esta pueda ser revisada por los evaluadores.	Hecho	21	2	MUST
HU-003	Yo, como administrador, necesito crear los usuarios del sistema, con la finalidad de guardar nuevos usuarios, como administradores, estudiantes y docentes.	Planificada	5	3	MUST
HU-004	Yo, como administrador, necesito editar los usuarios del sistema, con la finalidad de cambiar el rol y la información de los usuarios.	Planificada	5	3	MUST
HU-005	Yo, como administrador, necesito eliminar los usuarios del sistema, con la finalidad de que estos sean dados de baja.	Planificada	3	4	MUST
HU-006	Yo, como administrador, necesito listar los usuarios del sistema, con la finalidad de encontrar los usuarios registrados en el sistema y poder gestionarlos	Planificada	3	3	MUST
HU-007	Yo, como usuario, necesito poder autenticarme en el sistema, con la finalidad de poder navegar en todas las opciones disponibles para mi rol.	Planificada	8	5	MUST
HU-008	Yo, como director, necesito evaluar el proyecto de los estudiantes, con la finalidad de que la ficha quede revisada y con observaciones para que pueda ser evaluada	Planificada	5	3	MUST
HU-010	Yo, como estudiante, necesito, consultar las fichas de anteproyecto asociadas a los estudiantes autenticados en el sistema, con la finalidad de poder, conocer el detalle de la ficha de anteproyecto y poder editarla o eliminarla.	Hecho	3	2	MUST
HU-014	Yo, como estudiante, necesito editar la ficha de anteproyecto, con la finalidad de poder corregir y/o completar la información de la ficha.	Planificada	5	3	MUST
HU-015	Yo, como director de proyecto, necesito editar la ficha de anteproyecto, con la finalidad de poder asignar los docentes evaluadores.	Planificada	13	5	MUST
HU-016	Yo, como estudiante, necesito eliminar una ficha creada, con la finalidad de poder dar de baja a una ficha en el sistema	Hecho	3	2	MUST
HU-017	Yo, como director del programa, necesito crear una nueva evaluación del proyecto, con la finalidad de que se pueda realizar la evaluación cuantitativa del proyecto de	Planificada	13	6	MUST
HU-018	Yo, como director de proyecto ó docente, necesito listar, editar las fichas de anteproyecto ó editar la evaluación, con la finalidad de ver las fichas de	Planificada	21	4	MUST
HU-019	Yo, como analista de desarrollo, necesito listar montar un arquetipo, con la finalidad de que la tecnología a implementar se integre y poder empezar a desarrollar.	Hecho	21	1	MUST
HU-002	Yo, como jurado, necesito evaluar el proyecto de los estudiantes, con la finalidad de que el proyecto sea evaluado y pueda finalizarse.	Planificada	21	7	SHOULD
HU-009	Yo, como director del programa, necesito listar, editar y/o anular las fichas de anteproyecto, con la finalidad de ver las fichas de anteproyectos creadas en el	Planificada	13	6	SHOULD
HU-013	Yo, como director del programa, necesito cerrar la ficha de anteproyecto, con la finalidad de que la ficha no se pueda editar.	Planificada	5	8	SHOULD
HU-011	Yo, como director del programa, necesito generar un reporte de la ficha, con la finalidad de, conocer la información de las fichas y sus estados y hacer análisis de la información.	Planificada	13	8	COULD
HU-012	Yo, como docente, necesito recibir un correo de notificación, con la finalidad de saber de que fui asignado como director de un proyecto.	Planificada	5	8	COULD

Los puntos que se utilizan para la estimación es la presentada por la técnica póker utilizando la secuencia Fibonacci 1, 2, 3, 5, 8, 13, 21, 34, 55

Ahora, se realiza un Sprint Planning donde se crean los respectivos Sprints Backlogs que contienen los requisitos que deben realizarse por cada Sprint.

En el Sprint Planning, se determina que se van a realizar 8 sprints, sumando una complejidad de 186 puntos de esfuerzo.

Cada sprint será de dos semanas, con un compromiso promedio de 23 puntos por cada sprint.

6.2.2.5 Sprint 5

Se van a realizar las historias HU-007 y HU-015 con un total de 21 puntos de esfuerzo.

6.2.2.6 Sprint 6

Se van a realizar las historias HU-017 y HU-009 con un total de 26 puntos de esfuerzo.

6.2.2.7 Sprint 7

Se va a realizar la historia HU-002 con un total de 21 puntos de esfuerzo.

6.2.2.8 Sprint 8

Se van a realizar las historias HU-013, HU-011 y HU-012 con un total de 23 puntos de esfuerzo.

6.3 Implementar

6.3.1 Aplicación de la técnica TDD

Para empezar a implementar la técnica TDD, se procede a instalar el PHPUnit en su versión 5.7.27. PHPUnit es un entorno para realizar pruebas unitarias en el lenguaje de programación PHP.

Abrimos la consola y nos dirigimos a la ruta del proyecto phpUnit en C:\xampp\htdocs y ejecutamos la carpeta de pruebas del proyecto. `PHPUNIT ../../TEST` ya que se corre desde la carpeta vendor/bin

Figura 7 Ejecución de los tests

```
C:\xampp\htdocs\proyectoplus\vendor\bin
λ phpunit --colors ../../tests|
```

Se ejecuta el comando para probar la suite de pruebas que se va a encontrar ubicada en la carpeta Tests.

La carpeta de los Tests se manejan por fuera de la estructura, la idea es manejarlo por separado.

Antes de escribir el primer test, se define que la estructura del test es la siguiente figura

Figura 8 Estructura del Test Unitario

```

3
4 namespace EjemplosNUnit
5 {
6     [TestFixture]
7     public class NameNormalizerTests
8     {
9         [Test]
10        public void FirstLetterUpperCase()
11        {
12            // Arrange
13            string name = "pablo_rodriguez";
14            NameNormalizer normalizer =
15                new NameNormalizer();
16            // Act
17            string result =
18                normalizer.FirstLetterUpperCase(
19                    name);
20            // Assert
21            Assert.AreEqual("Pablo_Rodriguez", result);
22        }
23    }
24 }
```

(2010). Diseño Ágil con TDD. [Figura].

El test consta de 3 partes, Arrange (Preparar), Act (Actuar) y Assert (Afirmary).

Una parte de la preparación contiene el método SetUp. Si la etapa de preparación es común para varios tests.

El acto es llamar al código que se quiere probar y la afirmación se hace sobre el resultado de la ejecución de dicho código que se va a probar.

Teniendo todos los aspectos anteriores y teniendo el PHPUnit, procedemos a escribir el primer test.

Para la implementación de la técnica TDD, se toma una Historia De Usuario, luego se toma un criterio de aceptación de esa historia de usuario y se procede a escribir el test unitario.

Se define una tabla To-Do en el cual se van a ir escribiendo las pruebas que se vienen a la mente, en el momento en que se esté escribiendo el código para solventar un test y en ese momento se nos ocurre una nueva prueba, esta será escrita en este listado, se le antepone el símbolo (+) para diferenciarlas de las otras pruebas.

Tabla 5 *Lista TO-DO de pruebas*

TO-DO
<p>Función Create retorna ok</p> <p>La ruta del recurso ficha/create debe existir.</p> <p>La vista debe contener un formulario con los campos título, planteamiento, formulación y un botón submit para guardar la información.</p> <p>El método Store guarda título planteamiento y formulación y retorna a una vista index.</p> <p>Index retorna un listado de fichas.</p> <p>+ La vista debe contener un formulario con los campos director, objetivo general, objetivos específicos, etc.</p> <p>+ El método Store guarda un registro en la base de datos.</p> <p>+ La búsqueda debe limpiarse para evitar inyección de código malicioso al hacer la consulta en la base de datos.</p>

En este formato se muestra como es la implementación de la técnica TDD con ánimo de ejemplificar como se utilizó el Test Driven Development.

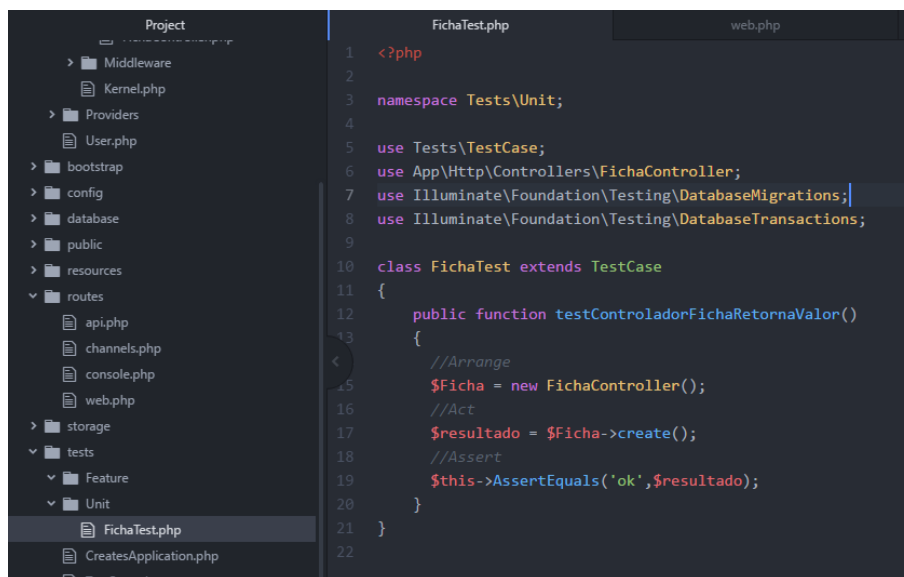
Tabla 6 Formato de aplicación de la técnica TDD 1

H. U.	HU-001
Descripción	Yo, como estudiante necesito crear la ficha de anteproyecto con la finalidad de que esta pueda ser revisada por los evaluadores.
TO-DO	Función Create retorna ok

Escribir una prueba que falle

Se procede a crear la prueba en la carpeta Tests del proyecto

Figura 9 Aplicación técnica TDD 1



Como se puede observar, aun no existe la clase FichaController que va a contener el método Create

Figura 10 Aplicación técnica TDD 2

```
C:\xampp\htdocs\projectoplus\vendor\bin
λ phpunit --colors ../tests
PHPUnit 5.7.27 by Sebastian Bergmann and contributors.

PHP Fatal error: Class 'App\Http\Controllers\FichaController' not found in C:\xampp
\htdocs\projectoplus\tests\Unit\FichaTest.php on line 27

Fatal error: Class 'App\Http\Controllers\FichaController' not found in C:\xampp\htdo
cs\projectoplus\tests\Unit\FichaTest.php on line 27
```

A esta parte del código donde la prueba genera un error, se le llama Rojo

Escribir un código que pase la prueba.

A continuación se escribe la clase FichaController

Figura 11 Aplicación técnica TDD 3

```
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class FichaController extends Controller
8 {
9
10 }
```

Se vuelve a ejecutar la prueba

Figura 12 Aplicación técnica TDD 4

```
PHPUnit 5.7.27 by Sebastian Bergmann and contributors.

E                                                                    1 / 1 (100%)

Time: 144 ms, Memory: 3.00MB

There was 1 error:

1) FichaTest::testExample
badMethodCallException: Method [create] does not exist.

C:\xampp\htdocs\proyectoplus\vendor\laravel\framework\src\Illuminate\Routing\Control
ler.php:82
C:\xampp\htdocs\proyectoplus\tests\Unit\FichaTest.php:30
C:\xampp\htdocs\proyectoplus\tests\Unit\FichaTest.php:30

ERRORS!
Tests: 1, Assertions: 0, Errors: 1.
```

Listo, ahora se procede a crear el método Create para que lograr que el código pase.

Figura 13 Aplicación técnica TDD 5

```
C:\xampp\htdocs\proyectoplus\vendor\bin
λ phpunit --colors ../tests
PHPUnit 5.7.27 by Sebastian Bergmann and contributors.

F                                                                    1 / 1 (100%)

Time: 684 ms, Memory: 3.25MB

There was 1 failure:

1) FichaTest::testExample
Failed asserting that null matches expected 'ok'.

C:\xampp\htdocs\proyectoplus\tests\Unit\FichaTest.php:33

FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

Se genera un error, ya que el método no está retornando ok. Se vuelve a modificar el método para que la prueba pase.

Figura 14 Aplicación técnica TDD 6

```
C:\xampp\htdocs\proyectoplus\vendor\bin
λ phpunit --colors ../tests
PHPUnit 5.7.27 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 146 ms, Memory: 3.00MB

OK (1 test, 1 assertion)
```

Queda en verde.

Refactorización del código.

Se busca que haya código duplicado, pero no encuentra, por ende no se realiza factorización en esta parte del ciclo TDD

Tabla 7 Formato de aplicación de la técnica TDD 2

H. U.	HU-010
Descripción	Yo, como estudiante necesito consultar las fichas de anteproyecto asociadas a los estudiantes autenticados en el sistema con la finalidad de poder conocer el detalle de la ficha de anteproyecto y poder gestionarlo.
TO-DO	<p>+ La búsqueda debe limpiarse para evitar inyección de código malicioso al hacer la consulta en la base de datos.</p> <p>Las siguientes cadenas de texto deben devolver los siguientes resultados:</p> <ul style="list-style-type: none"> • “cristian”, devuelve “cristian” • “cristian \$#%\$%&& botina”, devuelve “cristian botina” • “cristian 123 botina”, devuelve “cristian 123 botina” • “Jonh Doe' AND usuario is NULL; --”, devuelve Jhon Doe

- “loquesea' OR user LIKE 'a%';--”, devuelve Jhon Doe

Escribir una prueba que falle

Se escribe la prueba correspondiente, haciendo uso del **assertEquals** el cual realiza una comparación entre dos cadenas de texto, el primer argumento es la cadena esperada, simplemente es el valor correcto, como segundo argumento se recibe el valor actual de la cadena, que simplemente es el valor que retorna la función a probar y como tercer argumento se pasa una cadena con un mensaje lo cual va a mostrar en pantalla una vez que el test retorne un error por no encontrar igualdad en las dos cadenas comparadas.

Figura 15 Aplicación técnica TDD 7

```

118 public function testSanitizarVariablesDeBusqueda(){
119
120     $ficha = new FichaController();
121
122     $resultado = $ficha->sanearString('crístian');
123     $this->assertEquals('crístian', $resultado, "El valor no es el permitido");
124
125     $resultado = $ficha->sanearString('crístian $%$%&botina');
126     $this->assertEquals('crístian botina', $resultado, "El valor no es el permitido");
127
128     $resultado = $ficha->sanearString('crístian 123 botina');
129     $this->assertEquals('crístian 123 botina', $resultado, "El valor no es el permitido");
130
131     $resultado = $ficha->sanearString("John Doe' AND usuario is NULL");
132     $this->assertEquals('John Doe usuario is', $resultado, "El valor no es el permitido");
133
134     $resultado = $ficha->sanearString("John Doe' OR user LIKE 'a%';--");
135     $this->assertEquals('John Doeuser a', $resultado, "El valor no es el permitido");
136
137     $resultado = $ficha->sanearString("John Doe' or user like 'a%';--");
138     $this->assertEquals('John Doeuser a', $resultado, "El valor no es el permitido");
139
140     $resultado = $ficha->sanearString("CORTE");
141     $this->assertEquals('CORTE', $resultado, "El valor no es el permitido");
142
143 }
144

```

La prueba debe generar error para que se encuentre en rojo.

Figura 16 Aplicación técnica TDD 8

```

C:\xampp\htdocs\proyectoplus\vendor\bin
λ phpunit --colors ../../tests
PHPUnit 5.7.27 by Sebastian Bergmann and contributors.

.....E                                     7 / 7 (100%)

Time: 1.96 seconds, Memory: 12.00MB

There was 1 error:

1) Tests\Unit\FichaTest::testSanitizarVariablesDeBusqueda
BadMethodCallException: Method [sanearString] does not exist.

C:\xampp\htdocs\proyectoplus\vendor\laravel\framework\src\Illuminate\Routing\Controller.php:82
C:\xampp\htdocs\proyectoplus\tests\Unit\FichaTest.php:122
C:\xampp\htdocs\proyectoplus\tests\Unit\FichaTest.php:122

ERRORS!
Tests: 7, Assertions: 41, Errors: 1.

```

El primer error generado es por la ausencia de un método ubicado en la clase `fichaController` llamado `sanearString`

Escribir un código que pase la prueba.

Encontramos un error y es que la función `sanearString` no existe, por eso rápidamente se procede a escribir la función vacía, que retorne cualquier valor.

Figura 17 Aplicación técnica TDD 9

```

21
22 public function sanearString($cadena){
23     $cadena = $cadena;
24     return $cadena;
25 }
26
27

```

Al escribir la prueba, se procede a probar la función en el entorno PHPUnit. Se supera la prueba de la definición de la función, pero se generan otros errores de otros Asserts.

Figura 18 Aplicación técnica TDD 10

```

C:\xampp\htdocs\proyectoplus\vendor\bin
λ phpunit --colors ../../tests
PHPUnit 5.7.27 by Sebastian Bergmann and contributors.

.....F 7 / 7 (100%)

Time: 2 seconds, Memory: 12.00MB
There was 1 failure:

1) Tests\Unit\FichaTest::testSanitizarVariablesDeBusqueda:
El valor no es el permitido.
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'cristian botina'
+'cristian $#%$%&& botina'

C:\xampp\htdocs\proyectoplus\tests\Unit\FichaTest.php:126

FAILURES!
Tests: 7, Assertions: 43, Failures: 1.

```

Se continúa con el ciclo TDD, escribimos más código que supere el siguiente Assert, que en este caso no está quitando los caracteres extraños.

Se le ingresa a la función el código para reemplace los caracteres extraños.

Figura 19 Aplicación técnica TDD 11

```

161
162 //Esta parte se encarga de eliminar cualquier caracter extraño
163 $cadena = str_replace(
164     array("\\", "-", "g", "-", "~",
165         "#", "@", "|", "!", "\\",
166         ".", "$", "%", "&", "/",
167         "(", ")", "?", "*", "i",
168         "c", "[", "^", "<code>", "]",
169         "+", "}", "{", "--", "--",
170         ">", "< ", ";", ", ", ":",
171         ".", "'", $cadena);
172
173 return $cadena;
174 }
175

```

Se ejecuta la prueba.

Figura 20 Aplicación técnica TDD 12

```

.....F 7 / 7 (100%)
Time: 1.92 seconds, Memory: 12.00MB

There was 1 failure:

1) Tests\Unit\FichaTest::testSanitizarVariablesDeBusqueda
El valor no es el permitido
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'John Doe  usuario is'
+'John Doe AND usuario is NULL'

C:\xampp\htdocs\proyectoplus\tests\Unit\FichaTest.php:132

FAILURES!
Tests: 7, Assertions: 45, Failures: 1.

```

Quedan algunas sentencias SQL reservadas por depurar, aunque no es necesario profundizar en esto ya que el ORM Eloquent se encarga de manejar la inyección de SQL.

Figura 21 Aplicación técnica TDD 13

```

163 //Esta parte se encarga de eliminar cualquier caracter extraño
164 $cadena = str_replace(
165     array("\\", "'", '"', " ", "\n",
166         "\t", "@", "|", "!", "\\",
167         "\r", "$", "%", "&", "/",
168         "(", ")", "?", ":", "}", "{",
169         "<?", "[", "A", "<code>", "]",
170         "+", "]", "(", ":", ":", ":",
171         ">", "< ", ";", ":", ":",
172         ".", ":", $cadena);
173
174
175 //sentencias SQL
176 $cadena = str_ireplace(
177     array('ALTER TABLE', 'OR ', 'AND', 'BETWEEN', 'COUNT', 'NULL', 'CREATE SYNONYM', 'CREATE TABLE', 'CREATE VIEW', 'DELETE', 'DISTINCT', 'DROP',
178         '', $cadena);
179
180 return trim($cadena);
181 }
182

```

Se vuelve a ejecutar la prueba.

Figura 22 Aplicación técnica TDD 14

```

C:\xampp\htdocs\proyectoplus\vendor\bin
λ phpunit --colors ../../tests
PHPUnit 5.7.27 by Sebastian Bergmann and contributors.

..... 7 / 7 (100%)

Time: 1.87 seconds, Memory: 12.00MB

OK (7 tests, 48 assertions)

```

El test queda en verde.

Se hace llamado desde el index de la función, ya se comprobó que está funcionando bien, se puede hacer una pequeña prueba desde la aplicación.

Figura 23 Aplicación técnica TDD 15

```

26     public function index(Request $request)
27     {
28         $rol_id = 6;
29         $busqueda = $this->sanearString($request->get('titulo'));
30         $fichas = Ficha::filterAndPaginate($busqueda,$rol_id,4);
31         return view('ficha.index', compact('fichas'));
32     }

```

Al realizar la búsqueda, esta se ejecuta sin ningún problema.

Figura 24 Aplicación técnica TDD 16

Fichas

Titulo	Objetivo General	Director	Programa	Fecha de Creación	Acción
El titulito 123	desarrollar prototipos excelentes	BEATRIZ EUGENIA MARIN LOPEZ	INGENIERÍA DE SISTEMAS	2018-03-31 01:03:48	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Refactorización del código.

Se encuentra que la instancia del objeto FichaController se encuentra repetido, se asigna a una función que las agrupe, haciendo referencia al SetUp de los Tests.

Figura 25 Aplicación técnica TDD 17

```

public function testControladorFichaRetornaValor()
{
    //Arrange
    $Ficha = new FichaController();
    //Act
    $resultado = $Ficha->create();
    //Assert
    $this->AssertFalse(empty($resultado));
}

```

Figura 26 Aplicación técnica TDD 18

```

26 public function testSanitizarVariablesDeBusqueda(){
27     $Ficha = new FichaController();
28
29     $resultado = $Ficha->sanearString('cristian');
30     $this->assertEquals('cristian', $resultado, "El valor no es el permitido");
31
32     $resultado = $Ficha->sanearString('cristian $$$$botina');
33     $this->assertEquals('cristian botina', $resultado, "El valor no es el permitido");
34
35     $resultado = $Ficha->sanearString('cristian 123 botina');
36     $this->assertEquals('cristian 123 botina', $resultado, "El valor no es el permitido");
37
38     $resultado = $Ficha->sanearString("John Doe' AND usuario is NULL");
39     $this->assertEquals('John Doe  usuario is', $resultado, "El valor no es el permitido");
40
41     $resultado = $Ficha->sanearString("John Doe' OR user LIKE 'a%';--");
42     $this->assertEquals('John Doeuser  a', $resultado, "El valor no es el permitido");
43
44     $resultado = $Ficha->sanearString("John Doe' or user like 'a%';--");
45     $this->assertEquals('John Doeuser  a', $resultado, "El valor no es el permitido");
46
47     $resultado = $Ficha->sanearString("CORTE");
48     $this->assertEquals('CORTE', $resultado, "El valor no es el permitido");
49 }

```

Después de aplicar la refactorización queda la siguiente función `getFicha`, que se encarga de instanciar un objeto `Ficha` lo cual podrá ser llamado públicamente en la clase.

Figura 27 Aplicación técnica TDD 19

```

public function getFicha()
{
    $FichaObjeto = new FichaController();
    return $FichaObjeto;
}

```

Las líneas repetidas ahora llaman una función que retorna un objeto ficha, quedando de la siguiente forma:

Figura 28 Aplicación técnica TDD 20

```

27     public function testControladorFichaRetornaValor()
28     {
29         //Arrange
30         $Ficha = $this->getFicha();
31         //Act
32         $resultado = $Ficha->create();
33         //Assert
34         $this->AssertFalse(empty($resultado));
35     }
36

```

Figura 29 Aplicación técnica TDD 21

```

126     public function testSanitizarVariablesDeBusqueda(){
127         $Ficha = $this->getFicha();
128
129         $resultado = $Ficha->sanearString('cristian');
130         $this->assertEquals('cristian', $resultado, "El valor no es el permitido");
131
132         $resultado = $Ficha->sanearString('cristian $%$%&&botina');
133         $this->assertEquals('cristian botina', $resultado, "El valor no es el permitido");
134
135         $resultado = $Ficha->sanearString('cristian 123 botina');
136         $this->assertEquals('cristian 123 botina', $resultado, "El valor no es el permitido");
137
138         $resultado = $Ficha->sanearString("John Doe' AND usuario is NULL");
139         $this->assertEquals('John Doe  usuario is', $resultado, "El valor no es el permitido");
140
141         $resultado = $Ficha->sanearString("John Doe' OR user LIKE 'a%';--");
142         $this->assertEquals('John Doeuser  a', $resultado, "El valor no es el permitido");
143
144         $resultado = $Ficha->sanearString("John Doe' or user like 'a%';--");
145         $this->assertEquals('John Doeuser  a', $resultado, "El valor no es el permitido");
146
147         $resultado = $Ficha->sanearString("CORTE");
148         $this->assertEquals('CORTE', $resultado, "El valor no es el permitido");
149     }

```

Para ver el detalle la documentación, diríjase a carpeta ANEXOS\ DOCUMENTACION TECNICA TDD MEDIANTE PRUEBAS UNITARIAS PHPUNIT

6.3.2 Sprint 1

6.3.2.1 Sprint Backlog

Tabla 8 *Sprint Backlog Sprint 1*

Identificador (ID) de la Historia	Enunciado de la Historia	Estado	Estimación Poker	Iteración (Sprint)	Prioridad
HU-019	Yo, como analista de desarrollo, necesito listar montar un arquetipo, con la finalidad de que la tecnología a implementar se integre y poder empezar a desarrollar.	Planificada	21	1	MUST

Se plantea en este Sprint, poder montar toda la arquitectura necesaria para poder ejecutar la aplicación en las tecnologías seleccionadas, servidor, base de datos, entorno de ejecución de pruebas unitarias, etc.

El peso de este sprint es de 21 puntos de fuerza.

6.3.2.2 Log de Impedimentos

- No se puede ejecutar directamente el PHPUnit 7 directamente desde la carpeta que instala el laravel. genera problemas de incompatibilidad ya que el phpunit global instalado es el 5.7

6.3.2.3 Riesgos Identificados y Mitigados

- El problema de incompatibilidad de acuerdo a la versión del PHPUnit se puede consolidar como un riesgo que debe ser mitigado.
- Se mitiga, ejecutando la suite de pruebas en la versión 5.7 instalada en la carpeta vendor/bin del proyecto xampp/htdocs/proyectoplus/vendor/bin y ejecutando el PHPUnit con el siguiente comando `phpunit ../../tests`

6.3.3 Sprint 2

6.3.3.1 Sprint Backlog

Tabla 9 *Sprint Backlog Sprint 2*

Identificador (ID) de la Historia	Enunciado de la Historia	Estado	Estimación Poker	Iteración (Sprint)	Prioridad
HU-001	Yo, como estudiante, necesito crear la ficha de anteproyecto, con la finalidad de que esta pueda ser revisada por los evaluadores.	Hecho	21	2	MUST
HU-010	Yo, como estudiante, necesito, consultar las fichas de anteproyecto asociadas a los estudiantes autenticados en el sistema, con la finalidad de poder,	Hecho	3	2	MUST

	conocer el detalle de la ficha de anteproyecto y poder editarla o eliminarla.				
HU-016	Yo, como estudiante, necesito eliminar una ficha creada, con la finalidad de poder dar de baja a una ficha en el sistema	Hecho	3	2	MUST

Un gran objetivo importante planteado para el Sprint 1 es poder lograr la creación de una ficha por parte del estudiante, conllevando así también a la construcción del CRUD en Laravel aplicando la técnica TDD, lo cual servirá de experiencia en el manejo del framework y será muy útil para la implementación del resto de Sprints.

6.3.3.2 Log de Impedimentos

- Se considera un impedimento la falta de conocimiento en el framework.

6.3.3.3 Riesgos Identificados y Mitigados

- Se considera un riesgo el uso del framework Laravel por el simple hecho de que existe la falta de experiencia, pero se estima que este framework tiene un gran potencial para ser explotado, una vez se logre dominarlo sin dificultades.
- Se mitiga este riesgo, tomando un curso corto de Laravel en internet.

6.3.4 Sprint 3

6.3.4.1 Sprint Backlog

Tabla 10 *Sprint Backlog Sprint 3*

Identificador (ID) de la Historia	Enunciado de la Historia	Estado	Estimación Poker	Iteración (Sprint)	Prioridad
HU-003	Yo, como administrador, necesito crear los usuarios del sistema, con la finalidad de guardar nuevos usuarios, como administradores, estudiantes y docentes.	Planificada	5	3	MUST
HU-004	Yo, como administrador, necesito editar los usuarios del sistema, con la finalidad de cambiar el rol y la información de los usuarios.	Planificada	5	3	MUST

HU-006	Yo, como administrador, necesito listar los usuarios del sistema, con la finalidad encontrar los usuarios registrados en el sistema y poder gestionarlos	Planificada	3	3	MUST
HU-008	Yo, como docente necesito evaluar el proyecto de los estudiantes, con la finalidad de que la ficha quede revisada y con observaciones para que pueda ser evaluada por jurados.	Planificada	5	3	MUST
HU-014	Yo, como estudiante, necesito editar la ficha de anteproyecto, con la finalidad de	Planificada	5	3	MUST

poder corregir
y/o completar la
información de la
ficha.

6.3.4.2 Log de Impedimentos

- Aún sigue siendo un impedimento la falta de conocimiento en el framework.

6.3.4.3 Riesgos Identificados y Mitigados

- Se continúa con el curso de laravel, ya que este framework controla todos los aspectos del desarrollo en php.

6.3.5 Sprint 4

6.3.5.1 Sprint Backlog

Tabla 11 *Sprint Backlog Sprint 4*

Identificador (ID) de la Historia	Enunciado de la Historia	Estado	Estimación Poker	Iteración (Sprint)	Prioridad
HU-005	Yo, como administrador, necesito eliminar los usuarios del sistema, con la finalidad de que	Planificada	3	4	MUST

	estos sean dados de baja.				
HU-018	Yo, como director de proyecto o docente, necesito listar, editar las fichas de anteproyecto o editar la evaluación, con la finalidad de ver las fichas de anteproyectos creadas en el sistema, editarlas y/o editar la evaluación	Planificada	21	4	MUST

6.3.5.2 Log de Impedimentos

- No se registran logs de impedimentos, se va adquiriendo una curva de aprendizaje más alta, lo cual hace más fácil la implementación del sprint

6.3.5.3 Riesgos Identificados y Mitigados

- Con la experiencia que se va obteniendo del trabajo con el framewrok, no surgen nuevos riesgos, el desarrollo se va volviendo más rápido.

6.3.6 Sprint 5

6.3.6.1 Sprint Backlog

Tabla 12 *Sprint Backlog Sprint 5*

Identificador (ID) de la Historia	Enunciado de la Historia	Estado	Estimación Poker	Iteración (Sprint)	Prioridad
HU-007	Yo, como usuario, necesito poder autenticarme en el sistema, con la finalidad de poder navegar en todas las opciones disponibles para mi rol.	Planificada	8	5	MUST
HU-015	Yo, como director de proyecto, necesito editar la ficha de anteproyecto, con la finalidad de poder asignar los docentes evaluadores.	Planificada	13	5	MUST

6.3.6.2 Log de Impedimentos

- No se registran nuevos logs de impedimentos, la experiencia adquirida con el framework es más alta y no genera inconvenientes.

6.3.6.3 Riesgos Identificados y Mitigados

- No se registran nuevos riesgos en el desarrollo del Sprint 5

6.3.7 Sprint 6

6.3.7.1 Sprint Backlog

Tabla 13 *Sprint Backlog Sprint 6*

Identificador (ID) de la Historia	Enunciado de la Historia	Estado	Estimación Poker	Iteración (Sprint)	Prioridad
HU-009	Yo, como director del programa, necesito listar, editar y/o anular las fichas de anteproyecto, con la finalidad de ver las fichas de anteproyectos creadas en el	Planificada	13	6	SHOULD

	sistema, editarlas y/o anularlas.				
HU-017	Yo, como director del programa, necesito crear una nueva evaluación del proyecto, con la finalidad de que se pueda realizar la evaluación cuantitativa del proyecto de grado.	Planificada	13	6	MUST

6.3.7.2 Log de Impedimentos

- No se registraron nuevos logs de impedimentos en el sprint 7

6.3.7.3 Riesgos Identificados y Mitigados

- No se registran nuevos riesgos en el desarrollo del Sprint 5

6.3.8 Sprint 7

6.3.8.1 Sprint Backlog

Tabla 14 *Sprint Backlog Sprint 7*

Identificador (ID) de la Historia	Enunciado de la Historia	Estado	Estimación Poker	Iteración (Sprint)	Prioridad
HU-002	Yo, como jurado, necesito evaluar el proyecto de los estudiantes, con la finalidad de que el proyecto sea evaluado y pueda finalizarse.	Planificada	21	7	SHOULD

6.3.8.2 Log de Impedimentos

- No se registran nuevos logs de impedimentos en el sprint 7

6.3.8.3 Riesgos Identificados y Mitigados

- No se registran nuevos riesgos en el desarrollo del Sprint 5

6.3.9 Sprint 8

6.3.9.1 Sprint Backlog

Tabla 15 *Sprint Backlog Sprint 8*

Identificador (ID) de la Historia	Enunciado de la Historia	Estado	Estimación Poker	Iteración (Sprint)	Prioridad
HU-011	Yo, como director del programa, necesito generar un reporte de la ficha, con la finalidad de, conocer la información de las fichas y sus estados y hacer análisis de la información.	Planificada	13	8	COULD
HU-012	Yo, como docente, necesito recibir un correo de notificación, con la finalidad de saber de qué fui asignado como director de un proyecto.	Planificada	5	8	COULD
HU-013	Yo, como director del programa, necesito cerrar la	Planificada	5	8	SHOULD

ficha de
anteproyecto, con
la finalidad de
que la ficha no se
pueda editar.

6.3.9.2 Log de Impedimentos

- Se encuentra un impedimento para generar envío de correo de notificaciones por medio del servidor actual.
- Se debe buscar una librería compatible con el framework laravel para poder generar el documento PDF.

6.3.9.3 Riesgos Identificados y Mitigados

- El servidor debe contar con permisos para poder enviar correos.
- Se busca una librería para poder generar los reportes PDF, se encuentra la librería `barryvdh/laravel-dompdf`

6.4 Revisión y Retrospectiva

6.4.1 Revisión del sprint

Al finalizar cada Sprint, se programaron encuentros con el dueño del producto para revisar el desarrollo de las historias de usuario planeadas al principio de cada Sprint donde se validó que la entrega tuviera las características y funcionalidades definidas en las historias de usuario incluidas en el Sprint.

Adicional a lo anterior se ejecutaron las pruebas de caja negra y las pruebas unitarias automatizadas.

6.4.1.1 Pruebas.

6.4.1.1.1 Pruebas de Caja Negra

El sprint se valida con pruebas, para comprobar que cumple con los criterios de aceptación planteados en la historia de usuario, las pruebas implementadas son de caja negra, tomando como foco las entradas y salidas del sistema.

A continuación se listan las pruebas de caja negra generadas para las historias de usuario.

Tabla 16 *Listado de Casos de Pruebas*

Identificado r H.U.	Historia de Usuario	C.P. No.	Descripción Prueba
HU-001	Yo, como estudiante, necesito crear la ficha de anteproyecto, con la finalidad de que esta pueda ser revisada por los evaluadores.	CP-001	Verificar si se puede crear una ficha por un estudiante
		CP-002	Verificar si se cumple la excepción 1
		CP-003	Verificar si se cumple la excepción 2
		CP-004	Verificar si se cumple la excepción 3
HU-001	Yo, como estudiante, necesito crear la ficha de anteproyecto, con la finalidad de que esta	CP-005	Verificar si se puede crear una ficha por un estudiante

	pueda ser revisada por los evaluadores.		
HU-014	Yo, como estudiante, necesito editar la ficha de anteproyecto, con la finalidad de poder corregir y/o completar la información de la ficha.	CP-006	Verificar si se puede editar una ficha por un estudiante
		CP-007	Verificar si se cumple la excepción 1
		CP-008	Verificar si se cumple la excepción 2
		CP-009	Verificar si se cumple la excepción 3
		CP-010	Verificar si se puede editar una ficha por un estudiante
		CP-011	Verificar si se puede editar una ficha por un estudiante adicionando un objetivo.
		CP-012	Verificar si se cumple la excepción 4
HU-016	Yo, como estudiante, necesito eliminar una ficha creada, con la finalidad de poder dar de baja a una ficha en el sistema	CP-013	Verificar si se puede eliminar una ficha por un estudiante.
		CP-014	Verificar si se puede eliminar una ficha por un estudiante en un estado diferente a PENDIENTE
HU-015	Yo, como director de proyecto, necesito editar la ficha de anteproyecto, con la	CP-015	Verificar si se puede asignar un evaluador a la ficha
		CP-016	Verificar si se cumple la excepción 1

	finalidad de poder asignar los docentes evaluadores.	CP-017	Verificar si se cumple la excepción 2
		CP-018	Verificar si se cumple la excepción 3
HU-008	Yo, como docente necesito evaluar el proyecto de los estudiantes, con la finalidad de que la ficha quede revisada y con observaciones para que pueda ser evaluada por jurados.	CP-019	Verificar si se puede asignar una observación al objetivo estratégico de la ficha.
		CP-020	Verificar si se puede evaluar la ficha de anteproyecto
HU-017	Yo, como director del programa, necesito crear una nueva evaluación del proyecto, con la finalidad de que se pueda realizar la evaluación cuantitativa del proyecto de grado.	CP-021	Verificar si se puede asignar un jurado a la sustentación del proyecto de grado.
		CP-022	Verificar si se cumple la excepción 1
		CP-023	Verificar si se cumple la excepción 2
		CP-024	Verificar si se cumple la excepción 3
		CP-025	Verificar si se cumple la excepción 4
HU-002	Yo, como jurado, necesito evaluar el proyecto de los estudiantes, con la finalidad de que el	CP-026	Verificar si se puede evaluar la sustentación del proyecto de grado por un jurado.

	proyecto sea evaluado y pueda finalizarse.		
HU-018	Yo, como director de proyecto o docente, necesito listar, editar las fichas de anteproyecto o editar la evaluación, con la finalidad de ver las fichas de anteproyectos creadas en el sistema, editarlas y/o editar la evaluación.	CP-027	Verificar si se puede asignar o editar el director del proyecto
		CP-028	Verificar si cumple la excepción 1
HU-013	Yo, como director del programa, necesito cerrar la ficha de anteproyecto, con la finalidad de que la ficha no se pueda editar.	CP-029	Verificar si se puede dar visto bueno a la ficha
HU-007	Yo, como usuario, necesito poder autenticarme en el sistema, con la finalidad de poder navegar en todas las	CP-030	Verificar si se puede autenticar en el sistema
		CP-031	Verificar si se cumple la excepción 1

	opciones disponibles para mi rol.		
HU-003	Yo, como administrador, necesito crear los usuarios del sistema, con la finalidad de guardar nuevos usuarios, como administradores, estudiantes y docentes.	CP-032	Verificar si se puede crear un nuevo usuario
		CP-033	Verificar si se cumple la excepción 1
HU-004	Yo, como administrador, necesito editar los usuarios del sistema, con la finalidad de cambiar el rol y la información de los usuarios.	CP-034	Verificar si se puede editar un usuario en el sistema
		CP-035	Verificar si se cumple la excepción 1

Para ver el detalle de los casos de prueba, diríjase a carpeta ANEXOS\ CASOS DE PRUEBA

6.4.1.1.2 Pruebas Unitarias

Se realiza la implementación de la automatización de las pruebas unitarias, mediante PHPUnit, enfocándose en que cada prueba unitaria prueba ser ejecutada desde un entorno a través de una suite de pruebas. Ver el anexo de Desarrollo e implementación de la técnica TDD, ya que este implementa las pruebas unitarias.

6.4.2 Retrospectiva del sprint

Se realiza por cada Sprint una reunión con el dueño del producto para ver el incremento del producto como tal, se toman observaciones y sugerencias por el dueño del producto y las los avances de las historias de usuario son validadas con pruebas unitarias que vienen implícitos en el desarrollo de la técnica TDD y se realizan pruebas de caja negra donde se validan los aspectos de funcionalidad.

6.5 Lanzamiento

Se realiza la reunión con el dueño del producto para realizar la entrega del software web ProyectoPlus,

Después de que son aprobados los criterios de aceptación, se pone en marcha el lanzamiento en el ambiente de producción.

El despliegue se realiza remotamente por medio SSH en el servidor que se encuentra alojado en la universidad Antonio José Camacho.

Una vez se pone el software en un ambiente de producción, se procede con parametrizar las tablas maestras donde se guarda información de configuración.

Se libera la aplicación y se le dan credenciales a los usuarios finales que van a utilizar la aplicación web, luego se realiza una encuesta a los usuarios que manejan la aplicación para medir el nivel de satisfacción con la aplicación web

En los anexos se puede encontrar las encuestas realizadas a los usuarios

6.5.1 Encuesta

Se realiza una encuesta de satisfacción en la herramienta Google Forms para poder medir métricas de calidad en cuenta a la usabilidad.

Se busca medir las métricas de usabilidad planteadas en la ISO/IEC 25010 de la ISO 25000 Calidad de Producto de Software como:

Capacidad para reconocer su adecuación. Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.

Capacidad para ser usado. Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.

Protección contra errores de usuario. Capacidad del sistema para proteger a los usuarios de hacer errores.

Estética de la interfaz de usuario. Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.

Así pues, las preguntas que se plantearon para medir las métricas anteriores son las siguientes:

Figura 30 Resumen pregunta 1 del cuestionario de aceptación de la aplicación web ProyectoPlus

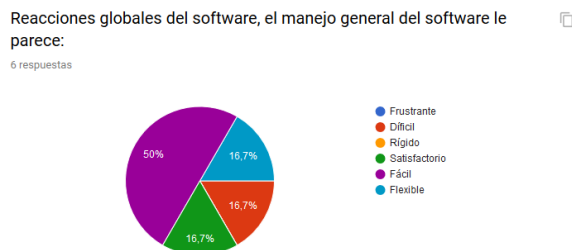


Figura 31 Resumen pregunta 2 del cuestionario de aceptación de la aplicación web
ProyectoPlus

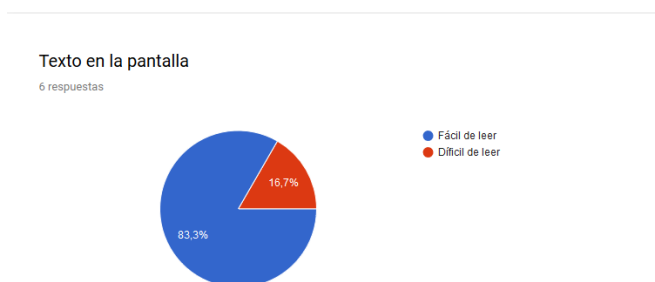


Figura 32 Resumen pregunta 3 del cuestionario de aceptación de la aplicación web
ProyectoPlus

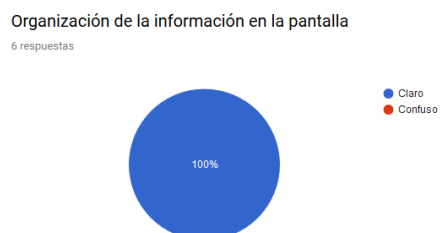


Figura 33 Resumen pregunta 4 del cuestionario de aceptación de la aplicación web
ProyectoPlus



Figura 34 Resumen pregunta 5 del cuestionario de aceptación de la aplicación web

ProyectoPlus

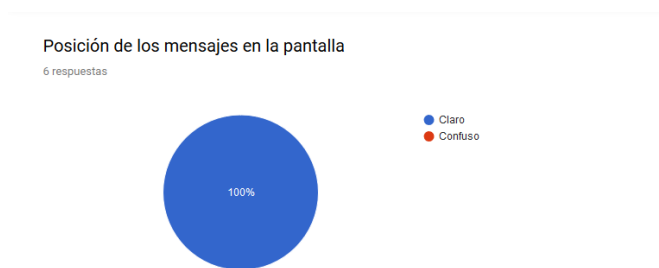


Figura 35 Resumen pregunta 6 del cuestionario de aceptación de la aplicación web

ProyectoPlus

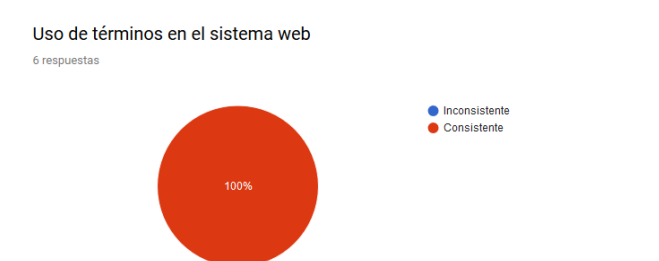


Figura 36 Resumen pregunta 7 del cuestionario de aceptación de la aplicación web

ProyectoPlus

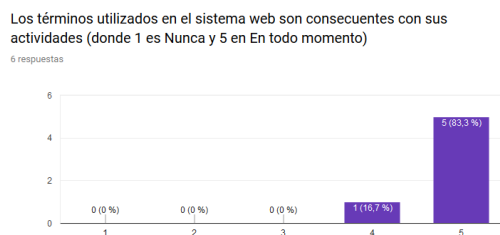


Figura 37 Resumen pregunta 8 del cuestionario de aceptación de la aplicación web

ProyectoPlus

Descripción de los mensajes en pantalla

6 respuestas



Figura 38 Resumen pregunta 9 del cuestionario de aceptación de la aplicación web

ProyectoPlus

Velocidad de la aplicación web (Donde 1 es Demasiado lento y 5 es Muy rápido)

6 respuestas

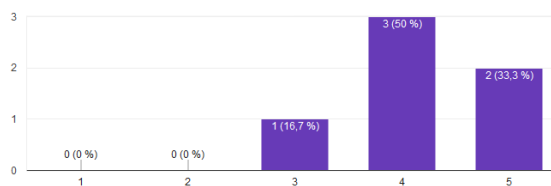


Figura 39 Resumen pregunta 10 del cuestionario de aceptación de la aplicación web

ProyectoPlus

Fiabilidad de la aplicación web (donde 1 es no fiable y 5 es muy fiable)

6 respuestas

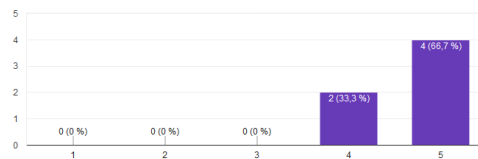
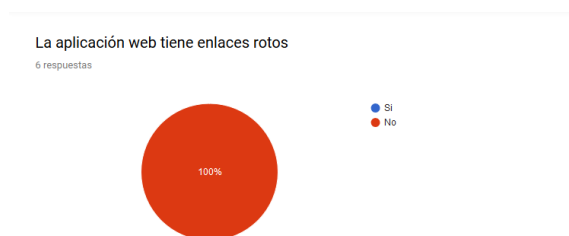


Figura 40 Resumen pregunta 11 del cuestionario de aceptación de la aplicación web
ProyectoPlus



6.6 Ventajas y Desventajas de Aplicar La Técnica TDD En La Metodología Scrum

A continuación las particulares de la aplicación de la técnica de desarrollo de software Test Driven Development se ven reflejadas en la siguiente comparativa realizada a partir de la experiencia adquirida al realizar la implementación del software de control de proyectos de grado con la técnica TDD y la metodología de desarrollo SCRUM.

Tabla 17 Comparación SCRUM con Híbrido Scrum + TDD

Características	Solo Scrum	Híbrido Scrum + TDD	Ventajas/Desventajas
Pruebas	Las pruebas se enfocan al final del desarrollo.	Se enfoca en la implementación del producto.	Las ventajas de usar TDD desde el mismo desarrollo, es que se a medida que se van escribiendo y desarrollando los tests unitarios, el diseño del código se va dando.
Tipos de pruebas frecuentes	Pruebas unitarias que se realizan manuales, si se implementan, se	Las pruebas deben ser automatizadas, para eso se usan	Garantiza que se van a detectar errores durante la misma

	usan al final del sprint.	diversos frameworks XUnit y se realizan durante cada sprint.	implementación del código permitiendo realizar el debido ajuste para solventar el error.
Curva de aprendizaje	Elementos de programación del producto	Conocimiento de herramientas de automatización, como el framework XUnit.	Esto es una desventaja para la aplicación de la técnica TDD en el proyecto, ya que se requiere contar con un nivel de conocimiento mayor en cuanto al lenguaje XUnit y la configuración de herramientas adicionales para realizar pruebas unitarias.
Documentación	Mínima	Exhaustiva	Es una desventaja ya que al tener un producto bajo el marco de TDD, requiere tener un poco más de documentación, ya que es vital que los nuevos programadores que llegan a colaborar con el desarrollo

			necesitan tener información vital del funcionamiento de la herramienta de automatización y la forma en que esta acoplado con el producto.
Estilo de procesos	Iterativo	Iterativo	Al aplicar la técnica en el proyecto, el estilo de procesos no cambia, ya que sigue siendo iterativo que se sigue representando en cada sprint
Calidad	Centrada en el Customer	Centrada en el customer y en las pruebas unitarias	Es una ventaja al presentar entregables que cumplen con los criterios de aceptación durante el desarrollo de cada sprint, ya que en cada sprint es debidamente testeado.

7. Diseño de Software

7.1 Arquitectura

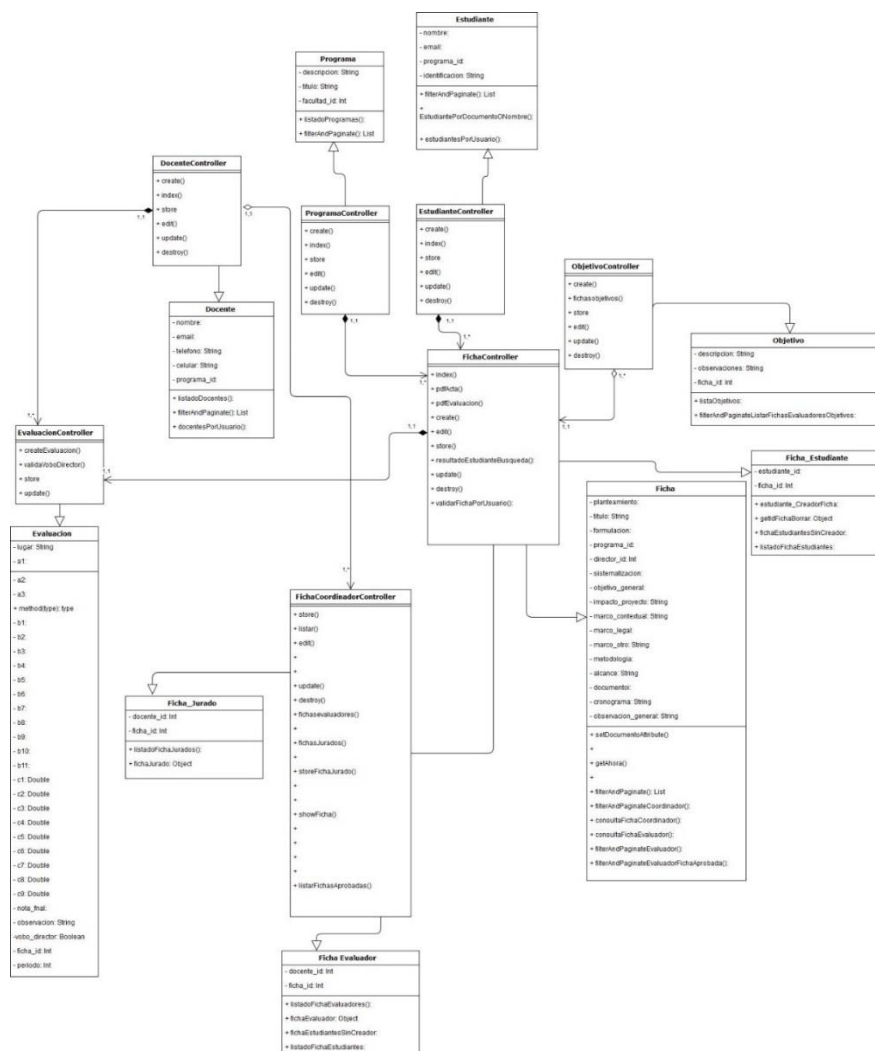
Para el desarrollo del sistema web para de proyectos para la Universidad Antonio José Camacho se define una arquitectura por capas básica Modelo Vista Controlador, añadiendo una capa para las rutas, ya que el framework Laravel utiliza este tipo de arquitectura.

El Framework Laravel se comunica por medio de esas capas, en el cual el controlador se comunica con las vistas y los modelos, las vistas son las encargadas de manejar el entorno gráfico y los modelos manejan la persistencia de los datos y por último la capa de las rutas establece la comunicación con los controladores.

7.1.1 Vista lógica y desarrollo

7.1.1.1 Diagrama De Clases

Figura 41 Diagrama de Clases

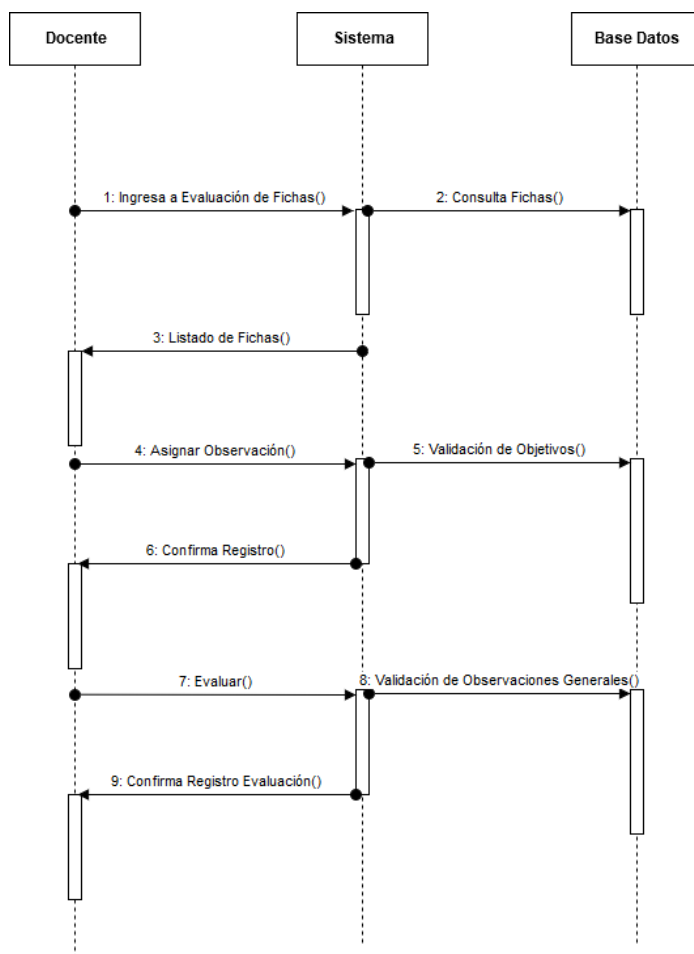


Para ver el detalle del diagrama de clases, diríjase a carpeta ANEXOS\ DISEÑO\VISTA LOGICA\DIAGRAMA DE CLASES\diagrama_clases

7.1.2 Vista de proceso

7.1.2.1 Vista de secuencia

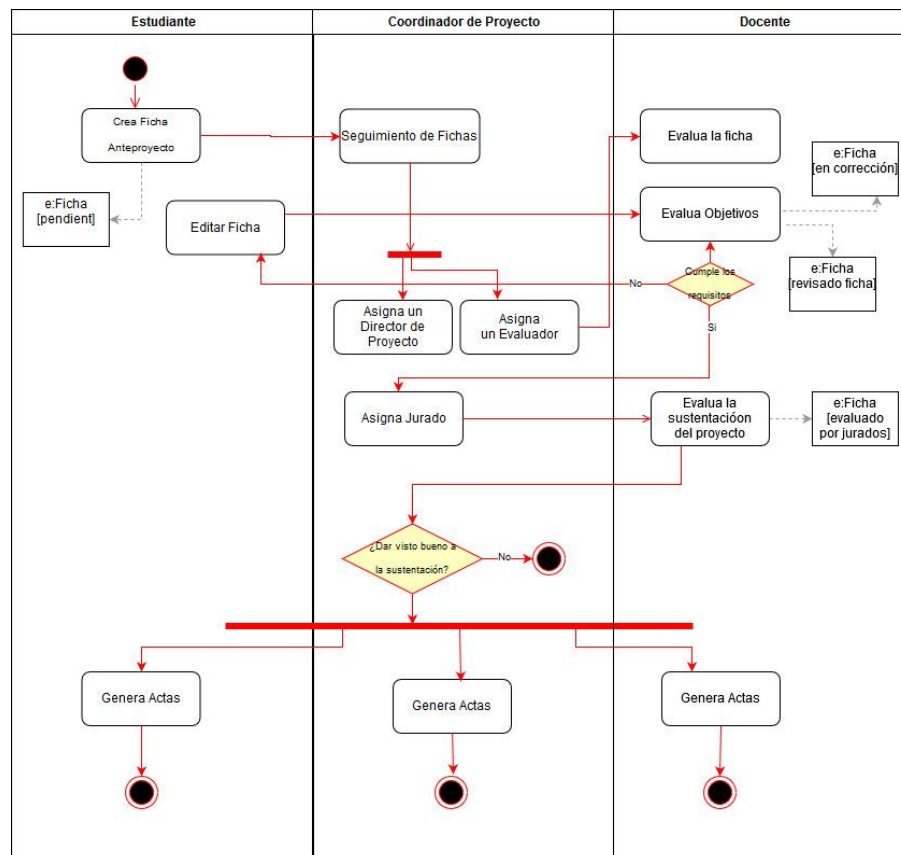
Figura 42 Diagrama de Secuencia Asignar Evaluador



Para ver el detalle de los diagramas de secuencia, diríjase a carpeta ANEXOS\
DISEÑO\VISTA PROCESO\DIAGRAMAS DE SECUENCIA

7.1.2.2 Diagrama de actividad

Figura 43 Diagrama de Actividades

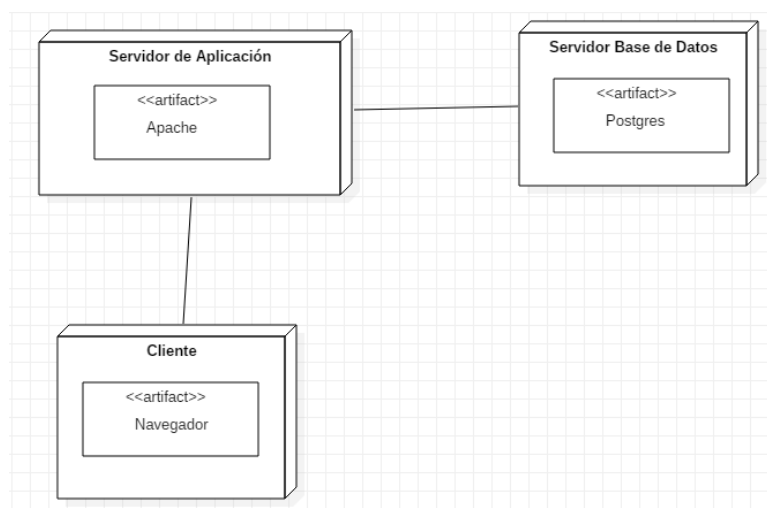


Para ver el detalle de los diagramas de secuencia, diríjase a carpeta ANEXOS\ DISEÑO\VISTA PROCESO\DIAGRAMA DE ACTIVIDADES\Diagrama Actividades

7.1.3 Vista física

7.1.3.1 Diagrama de despliegue

Figura 44 Diagrama de Actividades



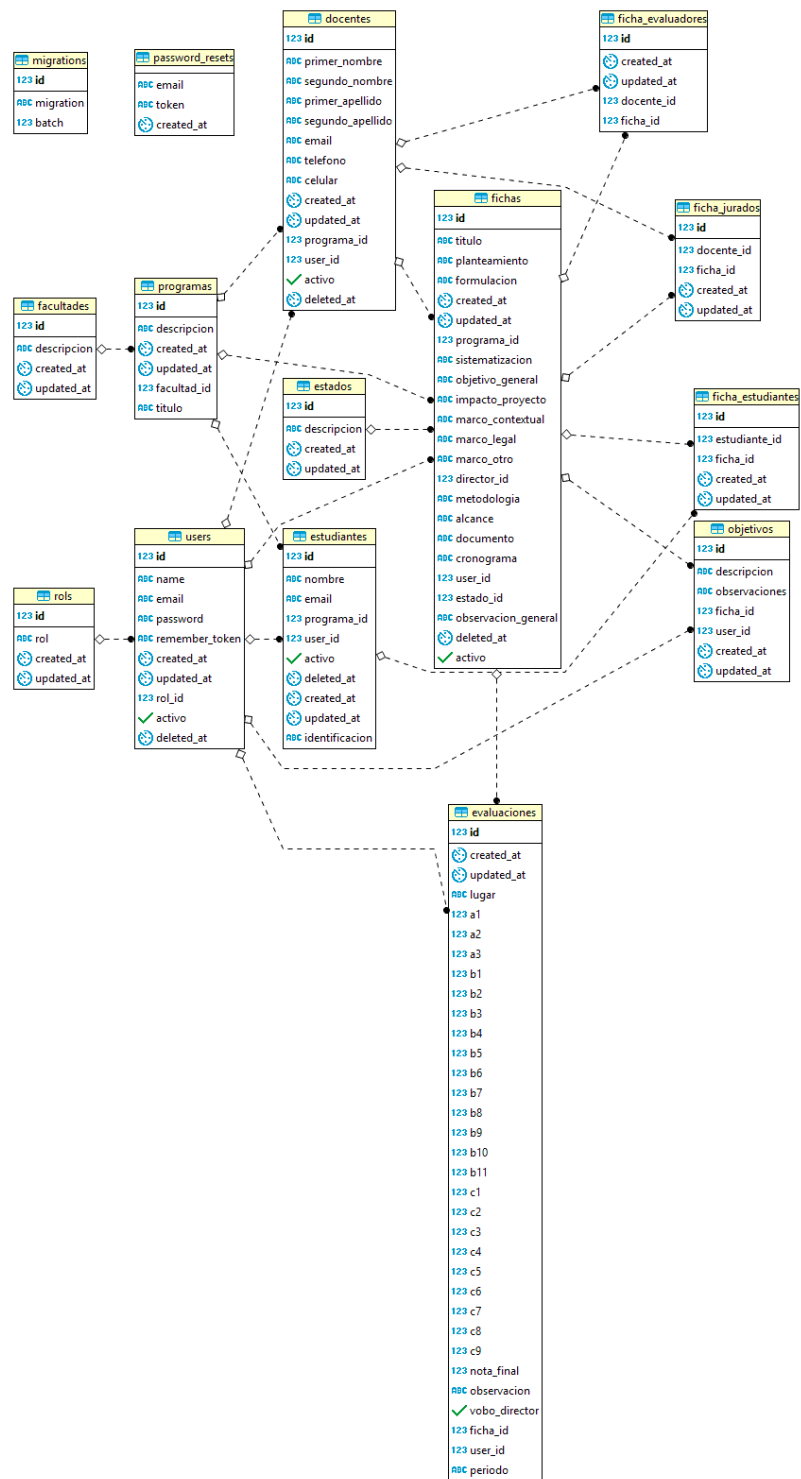
Para ver el detalle de los casos de prueba, diríjase a carpeta ANEXOS\ DISEÑO\VISTA FISICA\Diagrama Despliegue

7.2 Base de Datos

El motor seleccionado para guardar los datos fue Postgresql, en su versión 9.6, ya que este ya se encuentra funcionando en un entorno de producción para las aplicaciones de la universidad Antonio José Camacho, aparte de que es un motor de base de datos relacional, de licencia libre liberado bajo la licencia BSD y muy robusto.

7.2.1 Modelo Entidad Relación (MER)

Figura 45 Modelo de Entidad Relación (MER)



Para ver el detalle de los casos de prueba, diríjase a carpeta ANEXOS\ DISEÑO\MER\MER

8. Resultados

De acuerdo a la implementación de la técnica de desarrollo guiado por pruebas TDD, se obtiene como resultado con base a la experiencia obtenida que el producto de software está menos propenso a fallas como errores en funciones, errores en controladores o vistas, pero las fallas en funcionalidad quedan propensas a seguir dándose ya que estas se basan en la lógica del negocio, por eso es importante que los requisitos tomados en las reuniones siempre sean claros.

Así pues se logra obtener como resultados las ventajas y desventajas de la implementación de la técnica TDD en el proceso de desarrollo de este documento y la experiencia generada en la investigación de la técnica y la metodología Scrum.

Así mismo se logra documentar un proceso de desarrollo de un producto de software aplicando la técnica Test Driven Development junto con elementos de Scrum, obteniendo un software de gestión de proyectos de grado para la universidad Antonio José Camacho.

De igual manera la realización de este proyecto que fue documentado paso a paso sirve como material de apoyo para los desarrolladores de software que quieran implementar la técnica TDD combinada con una metodología ágil.

9. Conclusiones

Gracias a la experiencia obtenida por la aplicación de la técnica de Desarrollo guiado por pruebas TDD, se concluye que esta práctica de programación es una mejor alternativa cuando se quiere realizar un proyecto de software con alta calidad ya que esta provee un código más limpio

y fácil de leer por medio de la refactorización y uso de buenas prácticas. Al terminar el producto de software, generalmente ya está documentado como consecuencia del nombramiento de sus funciones permitiendo que cualquier programador pueda entender más fácil el código y la idea de negocio.

Conviene subrayar que la interacción de los estudiantes con la aplicación fue muy favorable ya que por medio de la encuesta que se realizó se pudieron determinar factores positivos para las métricas de usabilidad planteadas en la ISO/IEC 25010 de la ISO 25000.

Al proponer el desarrollo de un software aplicando la técnica TDD con estrategias ágiles de SCRUM, se pudo concluir que la técnica TDD no funciona sola. Esta técnica de desarrollo guiado por pruebas debe combinarse con una metodología que por lo general es XP ya que esta técnica fue incluida dentro de la metodología de Extreme Programming (Beck, Extreme Programming Explained: Embrace Change, 1999), pero para la aplicación de este estudio ha funcionado bien utilizando elementos de la metodología SCRUM.

Las ventajas que se pueden concluir con el desarrollo de esta aplicación de proyectos de grados son:

- Se da mayor calidad al producto de software, ya que las pruebas ejecutan fácil porque están automatizadas, evitando que las aplicaciones vayan a tener fallas.
- Se da más simplicidad al diseño ayudando a desarrollar programas robustos, fáciles de mantener y que se pueden modificar con sencillez.
- La Implementación de la técnica TDD produce un software menos propenso a fallas lo cual si se ve desde un punto de vista de financiero puede ayudar a disminuir costos gracias a que los programadores van a dedicar menos tiempo a apagar incendios después de haber lanzado el software y ese tiempo lo pueden usar desarrollando nuevos proyectos.

Las desventajas que se pueden concluir con el desarrollo de esta aplicación de proyectos de grados son:

- Es difícil realizar pruebas que necesitan datos de la base de datos haciendo que sea más demorado escribir pruebas que se encarguen de validar información en la capa de persistencia de datos.
- La aplicación de esta técnica es más conveniente en las funciones que retornan un resultado y reciben parámetros ya que la configuración de una prueba unitaria que reciba objetos instanciados de un formulario requiere más conocimiento del lenguaje en el que se programa haciendo más ardua la labor de probar un simple formulario, para esto más sencillo se necesitarían herramientas para realizar peticiones como Posman.
- La aplicación de la técnica TDD es más conveniente usar en la capa del negocio por ejemplo probar funciones que se encuentra dentro de los controladores ya que no se puede probar sobre una interfaz de usuario que tiene un formulario con botones, se requiere mayor configuración y mayor uso de tiempo generando pruebas para una interfaz.
- Es complejo implementar en vistas que están protegidas, ya que requiere estar autenticado en el sistema y el phpunit no tiene datos de acceso. Se deben centrar las pruebas en funciones que reciban parámetros y retorne variables.

Se puede concluir en general que actualmente se está prestando más atención a la calidad de software por medio de la aplicación de pruebas. Anteriormente las pruebas que se hacían era algo implícito que venía junto con la el desarrollo.

El enfoque que ofrece la técnica TDD de escribir primero las pruebas y luego escribir el código permite tener la seguridad de que la implementación no va a fallar, la automatización de las pruebas permite que cada vez que se haga un cambio el programador va a tener la seguridad de que el sistema seguirá comportándose de la misma forma.

10.Recomendaciones y Trabajos Futuros

Se identificaron necesidades futuras que requieren aplicar cambios en la aplicación para que esta pueda ser adaptada a las necesidades de otras facultades.

Se invita a continuar con el soporte de la aplicación y el desarrollo de nuevos módulos y funcionalidades que puedan mejorar significativamente el proceso de proyecto de grados de la Universidad Antonio José Camacho.

Se recomienda que la aplicación web siga siendo implementada con desarrollo guiado por pruebas y que se pueda complementar el uso de esta técnica de desarrollo con otras prácticas ágiles como ritmo sostenible, propiedad colectiva e integración continua.

11. Referencias Bibliográficas

- Astels, D. (2003). *Test-Driven Development: A Practical Guide: A Practical Guide*.
- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley.
- Beck, K. (2001). Aim, fire [test-first coding]. *IEEE Software*.
- Beck, K. (2002). *Test-Driven Development By Example*. Addison Wesley.
- Chaplin, D. (2003). Test First Programming. *Tech Zone*. Obtenido de <http://www.byte-vision.com/TestDrivenDevelopmentArticle.aspx>
- Clune, T., & Rood, R. (2011). Software testing and verification in climate model development. *IEEE Software*.
- Cuenca Girón, L. F., & Veintimilla, A. (s.f.). Estudio de la técnica Test Driven Development (TDD) y desarrollo del sistema para la administración de consultorios médicos. Obtenido de <http://repositorio.espe.edu.ec:8080/handle/21000/9094>
- Figueroa, R., Solís, C. J., & Cabrera, A. A. (2008). Metodologías tradicionales vs. Metodologías ágiles. Obtenido de <http://www.academia.edu/download/41231515/articulo-metodologia-de-sw-formato.doc>
- Fowler, M., & Beck, K. (1999). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley.
- Godoya, D. A., Kotynskib, H., Bellonic, E., Sosad, E., Gamonale, M., & Regueira, N. (2012). Simulando Proyectos de Desarrollo de Software que incorporan Test Driven Development. Obtenido de http://sedici.unlp.edu.ar/bitstream/handle/10915/54102/Documento_completo___.pdf-PDFA.pdf?sequence=1
- Guernsey, M. (2013). *Test-Driven Database Development: Unlocking Agility*. Addison-Wesley.
- Hendrickson, E. (2008). Driving Development with Tests. *Quality Tree Software*.
- Jurado, C. B. (2010). *Diseño Agil con TDD* (Primera ed.). SafeCREATIVE. Obtenido de http://www.carlosble.com/downloads/disenAgilConTdd_ebook.pdf
- Kanewala, U., & Bieman, J. (2014). Testing scientific software: A systematic literature review. *Information and Software Technology*, 14. Obtenido de www.elsevier.com/locate/infsof
- Lozano Moreno, C. (2015). Adaptación a PrestaShop de la tienda online rollandrollshop.com. Obtenido de

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/40002/5/clozanomoTFM0115memoria.pdf>

Meinke, K., & Niu, F. (2010). A learning-based approach to unit testing of numerical.

North, D. (2017). Introducing BDD. Obtenido de <http://dannorth.net/introducing-bdd>

Quingaluisa, A., Parra, R., & Gómez, E. (2015). Análisis de Test Driven Development y Scrum para el desarrollo rápido de aplicaciones basadas en Java y AngularJs. Obtenido de https://s3.amazonaws.com/academia.edu.documents/47162156/Analisis_de_Test_Driven_Development_y_Scrum_para_el_desarrollo_rapido_de_aplicaciones_basadas_en_Java_y_AngularJs.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1507664967&Signature=Rb1DTUMRVZav0d

SCRUMstudy. (2013). *Una guía para el CONOCIMIENTO DE SCRUM*. Phoenix: VMedu.

Shihab, E., Ming Jiang, Z., Adams, B., Hassan, A., & Bowerman, R. (2011). Prioritizing the creation of unit tests in legacy software systems.

Siniaalto, M. (2006). Test driven development: empirical body of evidence. *Information Technology for European Advancement*.

Vaca, P. A., Maldonado, C., Inchaurredo, C., Peretti, J., Romero, M. S., & Bueno, M. (2013). Test-Driven Development - Una aproximación para entender su utilidad en el proceso de desarrollo de Software. Obtenido de <http://www.conaiisi.unsl.edu.ar/portugues/2013/158-524-1-DR.pdf>

Zbucki, H., & Pons, C. (2015). Una Propuesta de Implementación MDD y TDD en el Dominio de Sistemas de Procesamiento Transaccional. Obtenido de http://sedici.unlp.edu.ar/bitstream/handle/10915/50318/Documento_completo.pdf-PDFA.pdf?sequence=1

12. Anexos

A continuación, se referencian los anexos que complementan la información mostrada en este documento, cuya raíz de directorio inicia con la ruta Documentación proyecto PROYECTOPLUS\ANEXOS:

12.1 Anexo 1

MANUAL DE USUARIO

12.2 Anexo 2

MANUAL DE INSTALACIÓN

12.3 Anexo 3

DOCUMENTACIÓN TÉCNICA TDD MEDIANTE PRUEBAS UNITARIAS PHPUNIT

12.4 Anexo 4

CASOS DE PRUEBA

12.5 Carpeta Anexo 5

Carpeta DISEÑO

Subcarpeta Anexo 5.1 DISEÑO

Subcarpeta 5.1.1 VISTA PROCESO

Subcarpeta 5.1.1.1 DIAGRAMA DE ACTIVIDADES

Subcarpeta 5.1.1.2 DIAGRAMA DE SECUENCIA

Subcarpeta 5.1.2 VISTA LÓGICA

Subcarpeta 5.1.2.1 DIAGRAMA DE CLASES

Subcarpeta 5.1.3 VISTA FÍSICA

Subcarpeta 5.1.4 MER

12.6 Anexo 6

Carpeta PLANEACIÓN

- Requerimientos Funcionales

- Requerimientos No Funcionales
- Product Backlog
- Prioritized Product Backlog
- Historias de Usuario
- Sprint Planning
- Listado Casos de Prueba

12.7 Anexo 7

CUESTIONARIO DE ACEPTACIÓN DE LA APLICACIÓN WEB PROYECTOPLUS V2

12.8 Anexo 8

RESULTADO CUESTIONARIO DE ACEPTACIÓN DE LA APLICACIÓN WEB
PROYECTOPLUS

12.9 Anexo 9

Carpeta BASE DE DATOS

Base_Datos.backup

12.10 Anexo 10

Carpeta APLICACIÓN

Proyectoplus.zip