**Dedicatoria**

A Dios porque sin él no estaría ni cerca de mis objetivos. A mis padres que siempre me impulsaron en mi formación y buenos valores. A mi hermano Hans por apoyarme como pocos lo harían.

**Agradecimientos**

A mi tutor Jimmy Villarroel por el tiempo dedicado, consejos y ayuda prestada al revisar los avances del proyecto. Al licenciado Rolando Jaldin R. por la paciencia e instrucción en estos últimos pasos de mi formación académica. A la licenciada Patricia Romero quien me preparo para concluir el documento y con todo el proceso en la elaboración del proyecto y por supuesto a todos los profesionales que tuve la oportunidad de tener como docentes, sin duda cada uno me dejó más de una enseñanza que no se limitó solo a una formación académica también me dejaron consejos para la vida que no olvidaré.

**Ficha resumen**

Índice de contenido

[**CAPÍTULO I INTRODUCCIÓN** 6](#_heading=h.4k668n3)

[1.1 Antecedentes 6](#_heading=h.2zbgiuw)

[1.2 Definición del problema 7](#_heading=h.1egqt2p)

[1.3 Objetivo general 7](#_heading=h.3ygebqi)

[1.4 Objetivos específicos 7](#_heading=h.2dlolyb)

[1.5 Área de conocimiento 7](#_heading=h.sqyw64)

[1.6 Alcance 7](#_heading=h.3cqmetx)

[1.7 Justificación 7](#_heading=h.1rvwp1q)

[**CAPÍTULO II MARCO TEÓRICO** 9](#_heading=h.4bvk7pj)

[**2.1 SISTEMAS DE INFORMACIÓN** 9](#_heading=h.2r0uhxc)

[2.1.1 Características 9](#_heading=h.1664s55)

[2.1.2 Elementos 10](#_heading=h.3q5sasy)

[**2.2 PROYECTO DE SOFTWARE** 10](#_heading=h.25b2l0r)

[**2.3 DESARROLLO DE SOFTWARE** 11](#_heading=h.kgcv8k)

[2.3.1 Gestión de proyectos 11](#_heading=h.34g0dwd)

[2.3.2 Planificación de proyectos de software 12](#_heading=h.1jlao46)

[2.3.3 Especificación del software 12](#_heading=h.43ky6rz)

[2.3.4 Diseño e implementación del software 13](#_heading=h.2iq8gzs)

[2.3.5 Validación del software 13](#_heading=h.xvir7l)

[2.3.6 Mantenimiento del software 13](#_heading=h.3hv69ve)

[**2.4 METODOLOGÍA SCRUM** 14](#_heading=h.1x0gk37)

[2.4.1 Características 14](#_heading=h.4h042r0)

[2.4.2 Proceso 16](#_heading=h.2w5ecyt)

[2.4.3 Ventajas 16](#_heading=h.1baon6m)

[**2.5 SISTEMAS PARA LA GESTIÓN ÁGIL DE PROYECTOS DE SOFTWARE** 16](#_heading=h.3vac5uf)

[2.5.1 Jira 16](#_heading=h.2afmg28)

[Herramientas para la planificación de sprints 17](#_heading=h.pkwqa1)

[2.5.2 Trello 17](#_heading=h.39kk8xu)

[Herramientas para la planificación 17](#_heading=h.1opuj5n)

[**2.6 LENGUAJE DE PROGRAMACIÓN PHP** 17](#_heading=h.48pi1tg)

[2.6.1 Características 18](#_heading=h.2nusc19)

[2.6.2 Ventajas 18](#_heading=h.1302m92)

[**2.7 ENTORNO DE DESARROLLO LARAVEL** 18](#_heading=h.3mzq4wv)

[2.7.1 Características 18](#_heading=h.2250f4o)

[2.7.2 Ventajas 19](#_heading=h.haapch)

[**2.8 ADMINISTRADOR DE BASE DE DATOS MYSQL** 19](#_heading=h.319y80a)

[2.8.1 Características 19](#_heading=h.1gf8i83)

[2.8.2 Ventajas 19](#_heading=h.40ew0vw)

[**CAPÍTULO III METODOLOGÍA DE DESARROLLO DEL SISTEMA** 21](#_heading=h.2fk6b3p)

[3.1 Product Backlog 21](#_heading=h.upglbi)

[3.2 Sprint backlog – Primer Sprint 21](#_heading=h.3ep43zb)

[3.2.1 Historias de usuario 22](#_heading=h.1tuee74)

[3.2.2 Prototipos de entrada y salida 23](#_heading=h.4du1wux)

[3.2.3 Modelo de datos 24](#_heading=h.2szc72q)

[3.2.4 Modelo físico de la base de datos 26](#_heading=h.184mhaj)

[3.2.5 Diseño test driven development 26](#_heading=h.3s49zyc)

[3.2.5 Implementación 27](#_heading=h.279ka65)

[3.2.6 Pruebas 29](#_heading=h.meukdy)

[3.3 Sprint backlog – Segundo Sprint 30](#_heading=h.36ei31r)

[3.3.1 Historia de usuario 30](#_heading=h.1ljsd9k)

[3.3.2 Prototipos de entrada y salida 32](#_heading=h.45jfvxd)

[3.3.3 Modelo de datos 32](#_heading=h.zu0gcz)

[3.3.4 Diseño TDD 33](#_heading=h.3jtnz0s)

[3.3.5 Implementación 34](#_heading=h.1yyy98l)

[3.3.6 Pruebas 34](#_heading=h.4iylrwe)

[**BIBLIOGRAFIA** 35](#_heading=h.2y3w247)

**ÍNDICE DE FIGURAS**

[Figura 1: Proceso de desarrollo de software 15](#_heading=h.35nkun2)

# **CAPÍTULO I INTRODUCCIÓN**

## 1.1 Antecedentes

El proceso de desarrollo de software tiene como objetivo final entregar un producto que cumpla con todos los requerimientos de un cliente. Sin embargo, su preocupación no es solamente esa, también tiene que enfocarse en los desafíos que puedan surgir en pleno proceso de desarrollo. “Actualmente, los negocios operan en un entorno global que cambia rápidamente. Tienen que responder a nuevas oportunidades y mercados” (Ian Sommerville, 2005, 258). A nivel mundial las deficiencias en todo tipo de proyectos son muchas y pueden determinar el éxito o fracaso de un proyecto, estas incluyen no realizar una buena organización y seguimiento del plan, además de otros factores como la capacitación del personal, los recursos disponibles, riesgos que puedan surgir a lo largo del proyecto, etc. Al no tener un plan estructurado para el desarrollo de software se incrementan las posibilidades de no terminar el proyecto a tiempo o incrementar los costos iniciales. La responsabilidad que se adquiere por tomar malas decisiones en el proceso de administración genera innumerables pérdidas económicas en todo el mundo.

Para facilitar la planificación de cualquier tipo de proyecto de software se tienen varias herramientas, que bien podrían ser las metodologías tradicionales, las cuales se quedaron en el pasado por su difícil adaptación al cambio o las metodologías ágiles que siguen incrementando su popularidad por su flexibilidad.

El software es parte de casi todas las operaciones de negocio, por lo que es fundamental que el software nuevo se desarrolle rápidamente para aprovechar nuevas oportunidades y responder a la presión competitiva. Por lo tanto, actualmente el desarrollo y entrega rápidos son a menudo los requerimientos más críticos de los sistemas software. (Ian Sommerville, 2005, 258).

Existen varias alternativas que ofrecen los servicios de gestión de proyectos, como Taiga y Trello, sin embargo, tienen ciertas restricciones económicas. Taiga pide por sus servicios 7$ mensuales por usuario, pudiendo rebajar a 5$ si se paga anualmente, mientras que Trello cobra 9.99$ por mes, considerando que estas opciones son las más baratas, el factor económico termina siendo un gran obstáculo para emplear estas herramientas ya sea por cualquier persona o incluso pequeñas organizaciones. También ofrecen administrar proyectos públicos ilimitados, pero con todo lo que significa tener algo tan delicado como la gestión de un proyecto a la vista de cualquiera que pueda encontrarlo puede traer muchos efectos negativos para la realización de este.

Un sistema web es una aplicación que se puede utilizar accediendo a un servidor web a través de internet mediante un navegador. Otorga varias ventajas como facilidad de uso, permite el trabajo colaborativo a distancia, son escalables, de rápida actualización y generan menos errores y problemas. Para el sistema propuesto, se pretende redactar una descripción para cada requerimiento, permitir un seguimiento a todos los ítems que se planifican en la lista de requerimientos para el desarrollo de un proyecto de software. El software implementa un seguimiento sencillo para cada uno de los requerimientos a realizar, los que están en desarrollo y los terminados, incluirá el encargado de realizar la tarea, también se podrá seleccionar una prioridad que podría ser alta, media o baja. Algunos de los beneficios que se obtendrían al utilizar el administrador de proyectos son: observar el avance del proyecto, además de ser una herramienta para permitir la comunicación entre el equipo de desarrollo acerca del progreso o problemas actuales, adicionalmente se permitirá compartir sugerencias entre los miembros para mejorar la calidad de cada tarea definida.

## 1.2 Definición del problema

Deficiente administración de los requerimientos en los proyectos de software por poco seguimiento del avance desarrollado.

## 1.3 Objetivo general

Proporcionar a las personas que desarrollan software una herramienta que les permita administrar el seguimiento a los requerimientos en sus proyectos, a través de un sistema web basado en principios de la metodología Scrum.

## 1.4 Objetivos específicos

* Permitir a los usuarios la creación y administración de varios proyectos al mismo tiempo.
* Proporcionar un taskboard por proyecto para aportar a la comunicación y refinamiento para cada requerimiento diseñado.
* Permitir elaborar los requerimientos o historias del product backlog.
* Facilitar la información sobre los avances en los requerimientos entre el equipo de desarrollo.

## 1.5 Área de conocimiento

Área: Sistemas de información

Sub- área: Sistemas web

## 1.6 Alcance

La plataforma no contempla almacenar código fuente para cada ítem del product backlog.

Se implementarán algunos conceptos de la metodología Scrum como ser historias de usuario, taskboard, estimación del esfuerzo y product backlog.

Permitirá observar el avance desarrollado para cada ítem del product backlog, una descripción de este y en qué estado se encuentra.

Facilitará el refinamiento de los requerimientos a través de comentarios redactados por los miembros del equipo de desarrollo.

Proveerá un control sobre cada requerimiento para asignarlo a un miembro del equipo, además cualquiera podrá revisarlo, también editarlo y marcarlo como terminarlo.

La plataforma será desarrollada para navegador web.

## 1.7 Justificación

Existen varios sitios web que ofrecen los servicios para la gestión de proyectos, y cumplen con su objetivo de favorecer el trabajo en el desarrollo de software, sin embargo, no son gratuitos, este costo que es por miembro del equipo, puede llegar a ser elevado sobre todo si se considera la cantidad de miembros y el pago que se tendría que efectuar por año. Puesto que las pequeñas empresas o grupos de trabajo no tienen los recursos para adquirir estos servicios, no gozan del acceso a estas herramientas y quedan limitadas a la planificación realizada en reuniones o de alguna forma inefectiva que en el peor de los casos acaba incrementado el tiempo de entrega del software y afectando la calidad que puede decaer hasta extremos e incluso pueden surgir problemas de organización que podrían dirigir el proyecto al fracaso.

Todo proyecto de software desde el caso más común que podría costar algunos miles de dólares hasta proyectos muy elaborados de millones de dólares, cualquiera sea el tipo, necesitan una correcta administración del proyecto durante todo el tiempo que se necesite hasta entregar el sistema. Estas herramientas de administración no surgieron de la noche a la mañana, ya están siendo aplicadas muchos años, lo que indica que si son tan usadas actualmente es porque traen muchos beneficios y aportan herramientas para superar los desafíos a los que se enfrentan las empresas diariamente, sobre todo para aquellas que están muy comprometidas con la calidad de cada proyecto que desarrollan. Los usuarios que utilicen la plataforma desarrollada verán las mejoras en la gestión del proyecto, mejor control sobre cada requerimiento, lo que reduciría la posibilidad de diferencias en el funcionamiento de las especificaciones o si las hubiera podrían ser aclaradas más rápidamente. Siendo otro de los aspectos que se busca mejorar la comunicación dentro del equipo de desarrollo, facilitar la respuesta al siempre esperado e indeseado cambio en los requerimientos/funcionalidades, en resumen, otorgar algunas de las herramientas y ventajas que nos ofrece la metodología Scrum.

# **CAPÍTULO II MARCO TEÓRICO**

En este capítulo se describirán los sistemas de información y que los caracteriza, las etapas por las que pasa la información. Las fases del desarrollo de software y los beneficios de las herramientas de la metodología scrum para el desarrollo, algunas de sus principales características, como ser su fácil adaptación al cambio. También se detallará el lenguaje PHP, su framework más popular laravel, el gestor de base de datos MySQL que serán las herramientas utilizadas para la implementación del sistema, además cuáles son las ventajas que aportan para el proyecto en particular.

## **2.1 SISTEMAS DE INFORMACIÓN**

Para poder entender de mejor manera un sistema de información se consideran las siguientes definiciones:

“Es un sistema cuyo principal propósito es gestionar y dar acceso a una base de datos de información. Los conflictos en los sistemas de información incluyen seguridad, usabilidad, privacidad y mantenimiento de la integridad de los datos. (Ian Sommerville, 2009, p. 18)

La interpretación del autor enfatiza el acceso y control de la base de datos, asimismo considera importante mantener la integridad de los datos almacenados.

“Un sistema de información está integrado de una gran variedad de elementos que se interrelacionan entre sí con el fin de apoyar las actividades de una empresa o negocio.” (Luis Domínguez, 2012, p. 34)

Domínguez reconoce que exiten elementos que forman parte de un sistema de información, también menciona la integración entre estos.

“Un sistema de información (IS) está formado por todos los componentes que colaboran para procesar los datos y producir información.” (Effy Oz, 2008, p. 11)

Nuevamente se dice que un sistema de información está formado por algo en este caso componentes, ¿cuáles son los componentes?

Al contar con varios planteamientos sobre sistemas de información y complementando un poco los enfoques acerca de, cuáles son los componentes o elementos involucrados y cuáles son las actividades principales de un sistema de información, se lo puede describir como:

Un conjunto de procesos, personas, tecnología y datos integrados para recolectar, procesar, almacenar y proveer información importante para desempeñar correctamente actividades de una organización o personas.

Los sistemas de información se utilizan para las actividades, de gestión y toma de decisiones, de una empresa u organización, proporcionan a sus usuarios la información que necesitan para su manejo e interpretación. Estos sistemas son usados con el fin de competir, innovar y lograr los objetivos planteados.

### 2.1.1 Características

La característica principal de los sistemas de información consiste en recopilar todos los datos que por separado no significan nada, para después juntarlos y convertirlos en información. Al terminar estos procesos se dispone de la información almacenada, obteniéndose finalmente conocimiento como salida o resultado. Lo que terminará siendo fundamental para el manejo adecuado de las decisiones debido a que posibilita actuar de acuerdo a los resultados obtenidos.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

1. **Entrada de Información:** Es el proceso mediante el cual el sistema de información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas.
2. **Almacenamiento de información:** El almacenamiento de datos de manera ordenada es una de las actividades más importantes, esta información es guardada para ser usada posteriormente.
3. **Procesamiento de Información:** En esta actividad se realizan operaciones con los datos almacenados lo que permite observar y analizar su contenido.
4. **Salida de Información:** La salida es la capacidad de un sistema de información para proveer la información útil en formato de documentos o informes para los usuarios.

### 2.1.2 Elementos

Los sistemas de información deben poseer hardware, software, datos, procesos y personas además de que deben interactuar entre ellos con el propósito de alcanzar los objetivos planteados por su organización. Estos elementos se describen a continuación.

**Hardware**

Dentro la que se encuentra el hardware que se compone de la tecnología que permite el almacenamiento, dispositivos de comunicaciones, entrada y salida de datos.

**Software**

Son los programas destinados a recoger los datos, almacenarlos, procesarlos y analizarlos, generando conocimiento para el usuario final.

**Datos**

Cualquier tipo de información relevante, como ejemplo un solo dato no describe nada, pero asociado a otro puede dar más información por eso su importancia de almacenar los datos de forma ordenada.

**Personas**

Son el elemento principal que forma el sistema de información, ya sea ingresando datos o utilizando los datos que se generan, o en otras palabras las personas son el recurso humano que interactúa con el sistema.

**Procesos**

Se define proceso como una serie de pasos para alcanzar un objetivo. Al realizar la automatización de los procesos, se obtiene un mejor control y manejo de los mismos.

El desarrollo de cualquier sistema de información debe pasar por fases del desarrollo de software como cualquier tipo de proyecto de software, a continuación, se profundiza sobre proyecto de software.

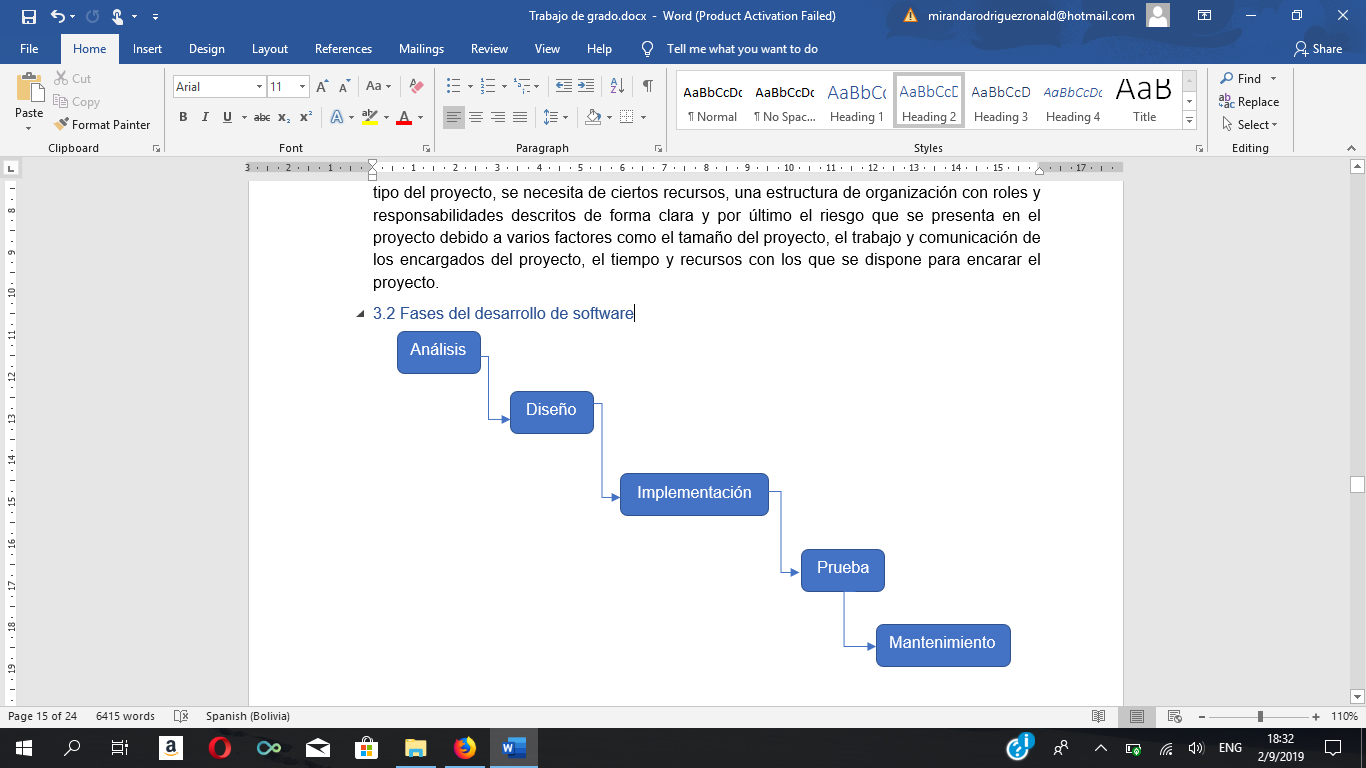
## **2.2 PROYECTO DE SOFTWARE**

Se puede considerar un proyecto de software como aquel que cuenta con las siguientes características. Tiene una serie de requerimientos que se pueden dividir en tareas, también que posee fechas de inicio y finalización determinadas al inicio del proyecto. Sin importar el tipo del proyecto, se necesita de ciertos recursos, una estructura de organización con roles y responsabilidades descritos de forma clara y por último el riesgo que se presenta en el proyecto debido a varios factores como el tamaño del proyecto, el trabajo y comunicación de los encargados del proyecto, el tiempo y recursos con los que se dispone para encarar el proyecto.

Por todo lo que implica el desarrollo de software durante la ejecución del proyecto, es recomendable dividir todo el trabajo que se debe realizar para facilitar los procesos, como respuesta a esta necesidad surgieron las fases de desarrollo de software que logran refinar considerablemente los procedimientos.

## **2.3 DESARROLLO DE SOFTWARE**

En la siguiente imagen se muestra las cinco etapas en el desarrollo de software tradicional análisis, diseño implementación, pruebas y mantenimiento.



*Figura 1: Proceso de desarrollo de software*

### 2.3.1 Gestión de proyectos

“La gestión de proyectos informáticos es un proceso que no se detiene hasta la finalización del proyecto. Para lograrlo se requiere de una estrategia global, junto con herramientas de trabajo que incrementen la productividad. El fin que se busca es planificar y controlar para obtener un plan para el desarrollo y administración de proyectos. Estos planes deben ayudar en los niveles estratégicos, tácticos y operaciones de las organizaciones para que logren sus objetivos planeados.” Guerra, L., y Bedini, A. (2005)

Durante el ciclo de vida de un proyecto, se tienen dos categorías de actividades a realizar y que además están relacionadas, estas son: actividades de gestión y actividades de desarrollo del sistema.

Las actividades de gestión son aquellas relacionadas con la administración de las organizaciones, personas, sistemas y procedimientos comprometidos en el proceso de planificación y construcción del sistema. Para la planificación del proyecto y las actividades de control, son iteradas para cada fase del proyecto y de esta manera proveen la estrategia de administración para que las actividades de desarrollo del sistema sean estimadas, programadas y ejecutadas.

Las actividades de desarrollo del sistema se enfocan en el desarrollo del proyecto. Generalmente están organizadas en distintas fases, que se agrupan en áreas funcionales de estudio, diseño y construcción basadas en una estructura de partición del trabajo a realizar.

### 2.3.2 Planificación de proyectos de software

La planificación requiere de objetivos que deben lograrse. Para los cuales se elaboran planes, procedimientos, establece una organización y asigna recursos y responsabilidades con el propósito de alcanzar objetivos propuestos. Y se obtiene como resultado la planificación el plan del proyecto.

**Objetivos de la Planificación de Proyectos de Software**

El principal objetivo de la planificación en proyectos de desarrollo de software es ordenar el qué hacer durante el proyecto y asignar adecuadamente los recursos y tareas para cumplir los objetivos propuestos.

En general se planifica para:

* Organizar el qué hacer del proceso de desarrollo de software.
* Minimizar tiempo y costos involucrados.
* Maximizar el uso de recursos disponibles.
* Establecer hitos del proyecto.
* Medir el avance.
* Mejorar la comunicación.
* Obtener soporte técnico.

La planificación es una tarea que se desarrolla al inicio del proyecto, pero rige el resto de las fases. Una buena planificación inicial ayudará a que las metas propuestas se cumplan y que los eventuales inconvenientes sean abordados de mejor forma.

### 2.3.3 Especificación del software

En esta etapa se analizan, comprenden y determinan todos aquellos servicios que el proyecto debe proveer, además de que se deben fijar las restricciones de lo que se espera durante el desarrollo del proyecto. Esta etapa es fundamental porque si esta etapa se completa de la mejor manera entrando en detalle acerca de cada funcionamiento esperado, no se corre el riesgo de que en el transcurso del proyecto surjan malentendidos por la funcionalidad esperada por parte del grupo de trabajo, clientes o los usuarios. Es decir, se debe consensuar entre el grupo de trabajo y los dueños del proyecto para aclarar cada funcionalidad con el mayor detalle posible o hasta que quede libre de ambigüedades.

Para comprender cuales son los servicios que se espera que el software realice, es necesario responder la pregunta ¿qué es lo que de verdad necesita el cliente?, por lo que se debe tener el cuidado de no confundir los requerimientos, esto debido a que, si bien el cliente y los desarrolladores pueden hablar el mismo idioma, la diferencia de conocimiento existente de unos, por un lado, más técnico y vista desde la lógica computacional. Por otra parte, los clientes están más centrados en el funcionamiento y conocimiento del negocio. Dando lugar a un intercambio entre ambos puntos de vista, que inicialmente será un poco complicado de entender a la otra persona por lo que, con el tiempo e interacción entre ambas partes, se logrará reducir esa diferencia de conocimiento.

Para la ingeniería de requerimientos se tienen 4 fases principales:

1. Estudio de viabilidad. En esta fase se determina si las herramientas de software y hardware actuales son suficientes para satisfacer las necesidades del usuario. Este estudio definirá si el proyecto se puede realizar y si su desarrollo es posible con los recursos disponibles.
2. Obtención y análisis de requerimientos. Se obtienen los requerimientos utilizando algunos métodos como ser la observación de sistemas similares existentes, entrevistas, reuniones con los usuarios o la revisión de la forma cómo administran, organizan y planifican sus procesos diarios. Toda esta información recopilada debe ser tratada eliminando ambigüedades o redundancias para que esta tenga las condiciones de ser comprendida.
3. Especificación de requerimientos. Donde se tiene como resultado toda la información investigada en la fase de análisis para tener todos los requerimientos documentados.
4. Validación de requerimientos. Todo el conjunto de requerimientos debe ser validado para que sean completamente consistentes y puedan ser aceptados por los involucrados cuando se termine el software.

### 2.3.4 Diseño e implementación del software

Con los requerimientos definidos que representan que es lo que se debe realizar, se continúa con la fase de diseño en la cual se utilizaran herramientas para representar aquello que se especificó y se debe implementar en el sistema, los datos de entrada, interfaces. Como al inicio difícilmente se podrá proponer un diseño completo que llegue a ser el mismo al finalizar el proyecto. El software debe ser modular para que se pueda modificar con facilidad, pudiéndose obtener un nuevo diseño cada vez que se tenga una nueva versión. Es decir que el diseño no se quedara en su primera representación, sino que con el desarrollo y entendimiento del proyecto surgirán a la vista errores o carencias en las especificaciones por lo que el diseño podría modificarse constantemente, obteniendo gracias a la retroalimentación mayor refinación de las especificaciones de todos los requerimientos a implementarse.

Para la fase de desarrollo no se tiene una norma que indique cómo se debe realizar la implantación de todo lo terminado en la fase de diseño, siendo libre la elección para los programadores de por dónde quieren empezar y cómo prefieren definir las especificaciones desglosadas en la fase previa. En resumen, se procura llevar el diseño de la fase anterior a código fuente.

### 2.3.5 Validación del software

Esencialmente constituye demostrar que el sistema funcione en base a las especificaciones del usuario, para lograr que el software se encuentre libre de errores se pueden realizar varios tipos de pruebas, estas se realizan con datos de prueba o preferentemente datos proporcionados por el cliente para analizar si de verdad el software está desarrollado a la medida con las características que necesita el cliente. El proceso termina cuando el desarrollador y el cliente validan que el sistema presentado cumple con todos los requerimientos definidos por ambas partes.

### 2.3.6 Mantenimiento del software

Al momento de validar el software puede parecer que todas las tareas fueron acabadas y no se verá más el código que se realizó, sin embargo, el ciclo de vida de un proyecto de software no acaba allí. Se debe contemplar realizar mejoras continuamente o reparar defectos que pasaron la prueba de validación y fueron detectados después de entregar el proyecto terminado. Por lo que el software podrá ser modificado mientras siga en funcionamiento.

Para que el desarrollo de un proyecto de software concluya satisfactoriamente no solamente se necesita seguir sus fases con detenimiento, también se necesitan metodologías de desarrollo, estas podrían ser ágiles o tradiciones. Por las herramientas y facilidad de uso que provee se eligió Scrum para el desarrollo del sistema propuesto.

## **2.4 METODOLOGÍA SCRUM**

Scrum es un método para trabajar en equipo a partir de iteraciones o sprints. Scrum es una metodología ágil, por lo que su objetivo será controlar y planificar proyectos con un gran volumen de cambios de última hora, donde la incertidumbre sea elevada.

Se suele planificar por semanas. Al final de cada Sprint o iteración, se va revisando el trabajo validado de la anterior semana. En función de esto, se priorizan y planifican las actividades en las que invertirán los recursos en el siguiente Sprint.

La metodología Scrum se centra en ajustar sus resultados y responder a las exigencias reales y exactas del cliente. De ahí, que se vaya revisando cada entregable, ya que los requerimientos van variando a corto plazo. El tiempo mínimo para un Sprint es de una semana y el máximo es de cuatro semanas.

Entre las principales características de la metodología Scrum, destaca que es un desarrollo incremental en lugar de la clásica planificación del desarrollo completo de un producto o servicio. Sus equipos de trabajo se caracterizan por ser auto-organizados. Y se centra en el producto final, principalmente en la calidad de este. Además, en la metodología Scrum se solapan diferentes fases de desarrollo, en lugar de llevar a cabo una planificación secuencial o de cascada.

### 2.4.1 Características

El marco de trabajo Scrum consiste en los Equipos Scrum, sus roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso.

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo (*time-boxes*), de tal modo que todos tienen una duración máxima. Una vez que comienza un Sprint, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.

Valores de Scrum

Son necesarios cinco valores para potenciar el trabajo realizado con la metodología, estos valores pueden ser aprendidos, vividos e incorporados mientras se utiliza la metodología, estos valores son compromiso, coraje, foco, apertura y respeto.

El dominio de estos valores sin duda, mejorará cada aspecto en el alcance de los objetivos planteados al inicio del proyecto. Es vital el compromiso para encontrar los motivos para esforzarse en lograr cada meta establecida. El coraje para trabajar de la forma adecuada, sin tomar atajos, especialmente cuando lo más sencillo sería tomar un camino con menos esfuerzo, que traerá consecuencias o al momento de afrontar los desafíos en cualquier etapa. Enfocarse en lo que de verdad requiere nuestra atención durante el sprint. Apertura para recibir recomendaciones y observaciones a nuestro trabajo, también para sugerir en procura de mejorar la experiencia en el trabajo. Por último, el respeto que debe existir entre todos los involucrados para tener un ambiente idóneo para el trabajo.

Pilares de Scrum

Se consideran tres pilares fundamentales para la implementación de Scrum transparencia, inspección y adaptación.

Transparencia

Todos los aspectos relevantes para el proceso deben ser accesibles para todos los que involucrados. también estos aspectos deben estar completamente definidos, no debe existir duda alguna en su interpretación, en otras palabras, todos deben comprender exactamente qué es lo que se espera.

Inspección

Es importante la inspección de los artefactos de Scrum para constatar el avance realizado en el sprint, se realiza para descubrir si existen retrasos o contratiempos. El intervalo de tiempo entre estas inspecciones debe ser prudente, ni tan extenso como para que no se obvien los problemas si existiesen y tampoco tan breve como para perjudicar el normal desarrollo del proyecto.

Adaptación

En caso de que no se cumplan con la fecha de entrega, proceso o la calidad del producto es inaceptable, el proceso debe modificarse con el objetivo de cumplir con lo acordado. O siendo más breve aún, si se presentaran contratiempos se debe buscar la manera de encauzar el proyecto hacia lo estipulado.

Lista de Producto

La Lista de Producto también conocida como product backlog es una lista ordenada de todo lo que se conoce que es necesario en el producto. Los elementos que la componen tienen como atributos descripción, orden, estimación y valor.

Una Lista de Producto nunca se puede considerar terminada. El diseño más temprano de la misma solo refleja los requisitos mejor entendidos al principio. La Lista de Producto crece a medida que el producto y el entorno en el que se usará también lo hacen.

Lista de Pendientes del Sprint

La Lista de Pendientes del Sprint en inglés sprint backloges el conjunto de elementos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. La Lista de Pendientes del Sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento “Terminado”.

La Lista de Pendientes del Sprint hace visible todo el trabajo que el Equipo de Desarrollo identifica como necesario para alcanzar el Objetivo del Sprint.

Incremento

El Incremento es la suma de todos los elementos de la Lista de Producto completados durante un Sprint y el valor de los incrementos de todos los sprints anteriores. Al final de un Sprint el nuevo Incremento debe estar “Terminado”, lo cual significa que está en condiciones de ser utilizado y que cumple la Definición de “Terminado” del Equipo Scrum. Un incremento es un cuerpo de trabajo inspeccionable y terminado que respalda el empirismo al final del Sprint. El incremento es un paso hacia una visión o meta.

El Sprint

Se considera como una parte esencial de Scrum, es un intervalo de tiempo (*time-box*) de aproximadamente un mes, también puede ser menos durante el cual se crea un incremento de producto utilizable y potencialmente desplegable. Es más conveniente si la duración de los sprints es consistente a lo largo del desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint anterior.

Cada sprint tiene como elementos una meta, un diseño y un plan flexible que permitirá su realización, el trabajo que deben realizar los miembros y un producto como resultado. Al durar hasta un mes, ayudan a repetir constantemente la inspección y adaptación del progreso además de que limitan el tiempo perdido. En caso de que suceda un fracaso solo se perdería un mes.

### 2.4.2 Proceso

Durante cada sprint, el cual tiene una duración de aproximadamente un periodo entre una y cuatro semanas (la magnitud es definida por el equipo y debe ser lo más corta posible), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo conversa con el Product Owner buscando claridad y magnitud adecuadas para luego determinar la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

### 2.4.3 Ventajas

Scrum es una propuesta de gestión basada en la división del trabajo en iteraciones, es decir, fases con objetivos y tareas específicas lo que aporta los siguientes beneficios:

* Facilita la planificación del proyecto, si contamos con objetivos, nos proporciona un rumbo claro y más aún con tareas definidas al principio con las que será más fácil alcanzar los objetivos planteados.
* Scrum es muy fácil de entender y requiere poco esfuerzo para comenzar a utilizarlo.
* Flexibilidad y adaptación a los contextos.Se adapta a cualquier contexto, área o sector de la gestión. No es una técnica exclusiva de ninguna disciplina.
* Gestión sistemática de riesgos**.**Del mismo modo, los riesgos que pueden afectar a un proyecto son gestionados en el mismo momento de su aparición.

Para la realización del sistema se utilizarán algunas herramientas de sistemas web basados en Scrum que permiten la gestión de proyectos.

## **2.5 SISTEMAS PARA LA GESTIÓN ÁGIL DE PROYECTOS DE SOFTWARE**

Actualmente se disponen de varios sistemas que permiten la gestión ágil de proyectos de software. Jira y trello se posicionan entre los más populares, cuentan con herramientas de la metodología Scrum y con la ventaja de ser fáciles de entender. Por lo que serán descritas las funcionalidades más importantes que estos ofrecen para el presente trabajo.

### 2.5.1 Jira

Jira Software es una herramienta ágil de gestión de proyectos compatible con cualquier metodología ágil, ya sea Scrum, Kanban o una propia. Desde tableros hasta informes ágiles, puedes planificar, supervisar y gestionar todos los proyectos de desarrollo de software ágil con una sola herramienta. Elige una metodología para ver cómo Jira Software puede hacer que tu equipo publique software de calidad con mayor rapidez.

Jira es una herramienta que utiliza las metodologías ágiles, Kanban e inclusive permite personalizarla, sin embargo, nos enfocaremos en la metodología ágil Scrum, debido a la importancia para el proyecto de grado.

#### Herramientas para la planificación de sprints

Facilita la supervisión de las versiones, funcionalidades y el progreso realizado.

Limpieza del backlog, permite la corrección de prioridades de historias de usuario. También las historias de usuario se pueden reordenar con un simple arrastre a la posición que se desea.

Para la planificación de sprints se estima historias, ajusta el alcance de los sprints, permite ver la velocidad y corrige la prioridad de las incidencias en tiempo real, para que el resto del equipo esté al tanto de los cambios.

Puntos de historia, en esta sección, permite estimar, supervisar y crear informes de los puntos de historia para que el equipo de desarrollo tenga más precisión en la estimación de sus sprints, también permite personalizar el método de estimación.

Una de las funcionalidades más importantes, tablero de Scrum ofrece añadir epics, responsables o proyectos. Para que al final del sprint se reciba una retroalimentación de todo lo sucedió en la realización del proyecto.

Supervisión y gestión de sprints que permite el control de permisos de sprints para definir lo que los usuarios del proyecto pueden ver o hacer.

### 2.5.2 Trello

Trello permite trabajar de forma más colaborativa y ser más productivo. Las tarjetas, listas y tableros de Trello le permiten organizar y priorizar sus proyectos de forma divertida, flexible y provechosa.

Trello es un sistema colaborativo que permite trabajar con el equipo de desarrollo, tarjetas, listas y tableros que se pueden gestionar para organizar los proyectos de forma más agradable y sencilla.

#### Herramientas para la planificación

Trello permite acceder a la información con comentarios, archivos adjuntos, fechas de vencimientos y más. Permite la colaboración en proyectos de principio a fin.

Ofrece crear tableros para cualquier proyecto, asignarle un nombre y añadir a los miembros del equipo.

Permite crear pasos de flujo de trabajo personalizados, también permite crear tarjetas para cada tarea por realizar.

En cada tarjeta se pueden añadir datos, fechas de vencimiento, comentarios y más.

También permite desplazar las tarjetas de una lista a otra.

Con el fin de desarrollar el sistema web se utilizaron varias herramientas para su implementación entre ellas se utilizó el lenguaje php cuya información se tiene a continuación.

## **2.6 LENGUAJE DE PROGRAMACIÓN PHP**

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

### 2.6.1 Características

* Gran extensión de documentación.
* Permite programación orientada a objetos.
* PHP es un lenguaje de propósito general que se ejecuta en el lado del servidor.
* Sirve sobre todo para realizar sitios dinámicos, interactuando con HTML y con bases de datos.
* Es multiplataforma, se puede instalar en cualquier sistema, ya sea Windows, Linux o Mac.
* Tiene una curva de aprendizaje baja, ya que se aprenden los comandos rápidamente y su sintaxis es sencilla de utilizar.
* La interacción entre PHP y HTML es muy sencilla.
* Es un lenguaje muy sólido y maduro, lleva muchos años en el mercado y se ha ido perfeccionando. A partir de la versión 7 se han introducido grandes cambios que permite un trabajo más sólido.
* Se puede separar fácilmente la estructura de PHP de los proyectos en el modelo vista-controlador con el que trabajan los frameworks.
* Hay una gran demanda laboral, ya que es muy utilizado y está presente en la mayoría de las páginas web existentes.

### 2.6.2 Ventajas

* Permite dividir scripts en métodos, clases, etc. para hacer más ágil de cara al servidor el procesamiento de los datos.
* PHP es totalmente libre y multiplataforma, por lo que no importa que sistema operativo se utilice. Su desarrollo es posible en cualquier sitio (siempre y cuando la máquina sea capaz de ejecutar el código).

Para favorecer el desarrollo con el fin de lograr avances sin tanto tiempo invertido o con menos código del necesario se incluirá el framework laravel que pondrá un toque moderno a un lenguaje un poco veterano como PHP.

## **2.7 ENTORNO DE DESARROLLO LARAVEL**

Laravel es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

El objetivo de Laravel es el de ser un framework que permita el uso de una sintaxis refinada para crear código de forma sencilla, evitando el desorden en el código y permitiendo multitud de funcionalidades. Aprovecha mucho de las ventajas de otros frameworks y utiliza las características de las últimas versiones de PHP.

La mayor parte de su estructura está formada por dependencias, especialmente de Symfony, lo que implica que el desarrollo de Laravel dependa también del desarrollo de sus dependencias.

### 2.7.1 Características

* Posee un poderoso conjunto de librerías.
* ORM increíble.
* Fácil enrutamiento.
* Autenticación simple.
* Sistema de ruteo, también [RESTful](https://www.ecured.cu/index.php?title=RESTful&action=edit&redlink=1).
* Blade, Motor de plantillas.
* Peticiones [Fluent](https://www.ecured.cu/index.php?title=Fluent&action=edit&redlink=1).
* Eloquent [ORM](https://www.ecured.cu/index.php?title=ORM&action=edit&redlink=1).
* Basado en Composer.
* Soporte para el caché.
* Soporte para [MVC](https://www.ecured.cu/index.php?title=MVC&action=edit&redlink=1).
* Usa componentes de [Symfony](https://www.ecured.cu/Symfony).
* Adopta las especificaciones PSR-2 y PSR-4.
* Cuenta con un código modular y extensible por medio de un administrador de paquetes.
* Soporte robusto para manejo de bases de datos.

### 2.7.2 Ventajas

* La documentación existente en la red es buena y completa.
* La comunidad que apoya este proyecto es enorme y sigue creciendo.
* Facilidad de integración con otras aplicaciones (Google, servicios de mailing, ERP’s, etc.).

## **2.8 ADMINISTRADOR DE BASE DE DATOS MYSQL**

MySQL, es un sistema de gestión de base de datos relacional o SGBD que utiliza la estructura Structured Query Language (SQL). Este gestor de base de datos en multihilo y multiusuario, esto le permite ser utilizado por varias personas al mismo tiempo, e incluso, realizar varias consultas a la vez, lo que lo hace sumamente versátil. También permite la gestión de base de datos relacionales, es decir almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad.

Nació como una iniciativa de Software Libre y aún sigue ofreciéndose como tal, para usuarios particulares. Pero si se desea utilizarlo para promover datos en una empresa, se puede comprar una licencia, como un software propietario, que es autoría de la empresa patrocinante (Actualmente Oracle Corporation).

La mayor parte del código se encuentra escrito en lenguaje C/C++ y la sintaxis de su uso es bastante simple, lo que permite crear bases de datos simples o complejas con mucha facilidad. Además, es compatible con múltiples plataformas informáticas y ofrece una infinidad de aplicaciones que permiten acceder rápidamente a las sentencias del gestor de base de datos.

### 2.8.1 Características

MySQL es un gestor de base de datos bastante versátil, que cuenta con una gran cantidad de opciones de trabajo:

* Cuenta con la capacidad de realizar tareas multiprocesador, debido a que posee la opción de trabajo multihilo.
* Puede ingresar una enorme cantidad de datos por columna de trabajo.
* Cuenta con API’s disponibles para los principales lenguajes de programación que existen.
* Aplicación con una portabilidad sobresaliente.
* Capacidad de soportar hasta 32 índices de tablas diferentes.
* Estupendo nivel de seguridad que permite gestionar varios usuarios con login y contraseñas individuales.
* Multiplataforma.

### 2.8.2 Ventajas

* Su enorme documentación y su capacidad de realizar pruebas de rendimiento en tiempo real.
* También, posee una función de empaquetamiento y ofrece la posibilidad de trabajar con listas de gran tamaño.
* Su mantenimiento es bastante simple y rápido.

En este capítulo se explicará en detalle la gestión de proyectos informáticos haciendo énfasis en la planificación, resaltando la ventaja de tener un orden definido de lo que se debe hacer, los medios que aporten al éxito del proyecto y las fases del proceso de desarrollo. Entre las que están análisis, diseño, implementación, pruebas y mantenimiento, que son las más importantes para la planificación del proyecto, por último, se revisará las funcionalidades que ofrecen sistemas similares al propuesto para tener la base de los requerimientos necesarios en la implementación del sistema.

# **CAPÍTULO III METODOLOGÍA DE DESARROLLO DEL SISTEMA**

En este capítulo se describe las fases del desarrollo del sistema web, así también las herramientas que son parte de la metodología de desarrollo.

## 3.1 Product Backlog

El product backlog hace referencia a la lista de productos del proyecto, que son los requerimientos definidos para la conclusión del proyecto.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Descripción | Prioridad | Estimación de esfuerzo  (días) | Sprint | Pruebas | Notas |
| 1 | Registro de cuentas de los usuarios. | 1 | 2 | 1 |  |  |
| 2 | Registro de proyectos del usuario. | 2 | 6 | 1 |  |  |
| 3 | Control de miembros del proyecto. |  | 4 | 1 |  |  |
|  | El sistema validara cuentas de usuarios registrados. |  | 1 | 1 |  |  |
|  | Un administrador podrá controlar las cuentas de los usuarios. |  | 2 | 1 |  |  |
| 4 | Registro de sprints para un proyecto. |  | 2 | 2 |  |  |
| 5 | Registro de historias de usuario para un sprint. |  | 6 | 2 |  |  |
|  | El sistema web debe estar disponible desde un servidor. |  | 1 | 2 |  |  |
|  | Un usuario puede registrar tareas para cada historia de usuario. |  | 3 | 3 |  |  |
|  | Un usuario puede subir/descargar documentos de las historias de usuario. |  | 4 | 3 |  |  |
|  | Un usuario contara con un taskboard para las tareas de cada historia de usuario. |  | 6 | 3 |  |  |
|  | Permitir escribir comentarios para las historias definidas. |  | 3 | 4 |  |  |
|  | Enviar mensajes al realizar cambios y asignaciones en tareas e historias. |  | 3 | 4 |  |  |
|  | Ver modificaciones realizadas por los demás miembros |  |  | 4? |  |  |
|  | Barra de estado de los sprints ?? |  |  | 4? |  |  |

## 3.2 Sprint backlog – Primer Sprint

En el siguiente cuadro se tienen los requerimientos y sus respectivas tareas para cada ítem del product backlog pertenecientes al primer sprint.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requerimiento/Tarea | Estado | Estimación de (hrs) esfuerzo |
| 1 | Registro de cuentas de los usuarios   * Crear nuevo usuario * Modificar datos del usuario * Eliminar usuario registrado | Concluido  Concluido  Concluido | 5  4  4 |
| 2 | Registro de proyectos del usuario   * Crear nuevo proyecto * Modificar datos del proyecto * Eliminar proyecto registrado * Listar proyectos creados * Listar proyectos a los que fue invitado | Concluido  Concluido  Concluido  Concluido  Concluido | 6  5  3  3  3 |
| 3 | Control de miembros del proyecto   * Eliminar miembros del proyecto * Añadir miembros al proyecto * Listar miembros registrados | Concluido  Concluido  Concluido | 2  4  2 |
|  | El sistema validará cuentas de usuarios registrados.   * Enviar mensaje con el enlace de la página web para la confirmación. * Validar cuenta al sistema. | Concluido  Concluido | 6  2 |
|  | Un administrador podrá controlar las cuentas de los usuarios.   * Eliminar cuenta del sistema. * Validar cuenta al sistema. | Concluido  Concluido |  |

### 3.2.1 Historias de usuario

En las siguientes tablas se mostrarán las historias de usuario, que describen la funcionalidad esperada para cada elemento del product backlog.

|  |  |  |  |
| --- | --- | --- | --- |
| ID: 1 | Título: Registro de cuentas de los usuarios | | |
| Est. Esf.: | Prioridad: alta | Dependiente de: | |
| Descripción:   * Para el registro se contará con un formulario con los campos nombre, email, contraseña y repetir contraseña. * La cuenta de los usuarios registrados podrá eliminarse. * El usuario podrá modificar sus datos. | | | |
| Prueba de aceptación:   1. Se podrán registrar nuevos usuarios. 2. Los usuarios podrán modificar sus datos. 3. No se podrá registrar el mismo correo más de una vez. 4. Los usuarios podrán eliminar sus cuentas. | | | |
| Elaborado por:  Ronald Miranda | Responsable:  Ronald Miranda | Fecha de Elaboración:  28/08/2019 | Versión:  1 |

|  |  |  |  |
| --- | --- | --- | --- |
| ID: 2 | Título: Registro de proyectos | | |
| Est. Esf.: | Prioridad: media | Dependiente de: | |
| Descripción:   * Para el registro de proyectos se contará con los campos nombre y descripción. * El dueño podrá eliminar sus proyectos registrados. * Los datos del proyecto se podrán modificar. * Se listarán los proyectos creado por el dueño y a los que fue invitado | | | |
| Prueba de aceptación:   1. Se registran los datos llenados en el formulario de un nuevo proyecto. 2. No se permitirá registrar más de un proyecto con el mismo nombre. 3. Modificar datos de un proyecto registrado. 4. Eliminar el proyecto registrado por el usuario. 5. Observar los proyectos que registro el usuario y a los que fue invitado. | | | |
| Elaborado por:  Ronald Miranda | Responsable:  Ronald Miranda | Fecha de Elaboración:  28/08/2019 | Versión:  1 |

|  |  |  |  |
| --- | --- | --- | --- |
| ID: 3 | Título: Gestión de miembros del proyecto | | |
| Est. Esf.: | Prioridad: media | Dependiente de: | |
| Descripción:   * Para añadir miembros al proyecto se dispondrá de un campo para ingresar el correo del usuario que se quiera invitar. * Los miembros registrados podrán eliminarse. * Se podrán ver datos los miembros del proyecto. | | | |
| Prueba de aceptación:   1. Se observará en la lista de miembros al usuario añadido. 2. Se podrá eliminar a miembros del proyecto. 3. No se podrá añadir a un usuario más de una vez. 4. El usuario eliminado no podrá ver información asociada al proyecto. | | | |
| Elaborado por:  Ronald Miranda | Responsable:  Ronald Miranda | Fecha de Elaboración:  28/08/2019 | Versión:  1 |

|  |  |  |  |
| --- | --- | --- | --- |
| ID: 4 | Título: El sistema validará las cuentas de usuarios registrados. | | |
| Est. Esf.: | Prioridad: media | Dependiente de: | |
| Descripción:   * Al registrarse un usuario, se enviará un mensaje a su correo electrónico con un enlace para activar su cuenta. * El sistema activará la cuenta, previa validación del enlace. | | | |
| Prueba de aceptación:   1. Recibir el mensaje de confirmación en el correo. 2. El usuario validado podrá ingresar al sistema. | | | |
| Elaborado por:  Ronald Miranda | Responsable:  Ronald Miranda | Fecha de Elaboración:  28/08/2019 | Versión:  1 |

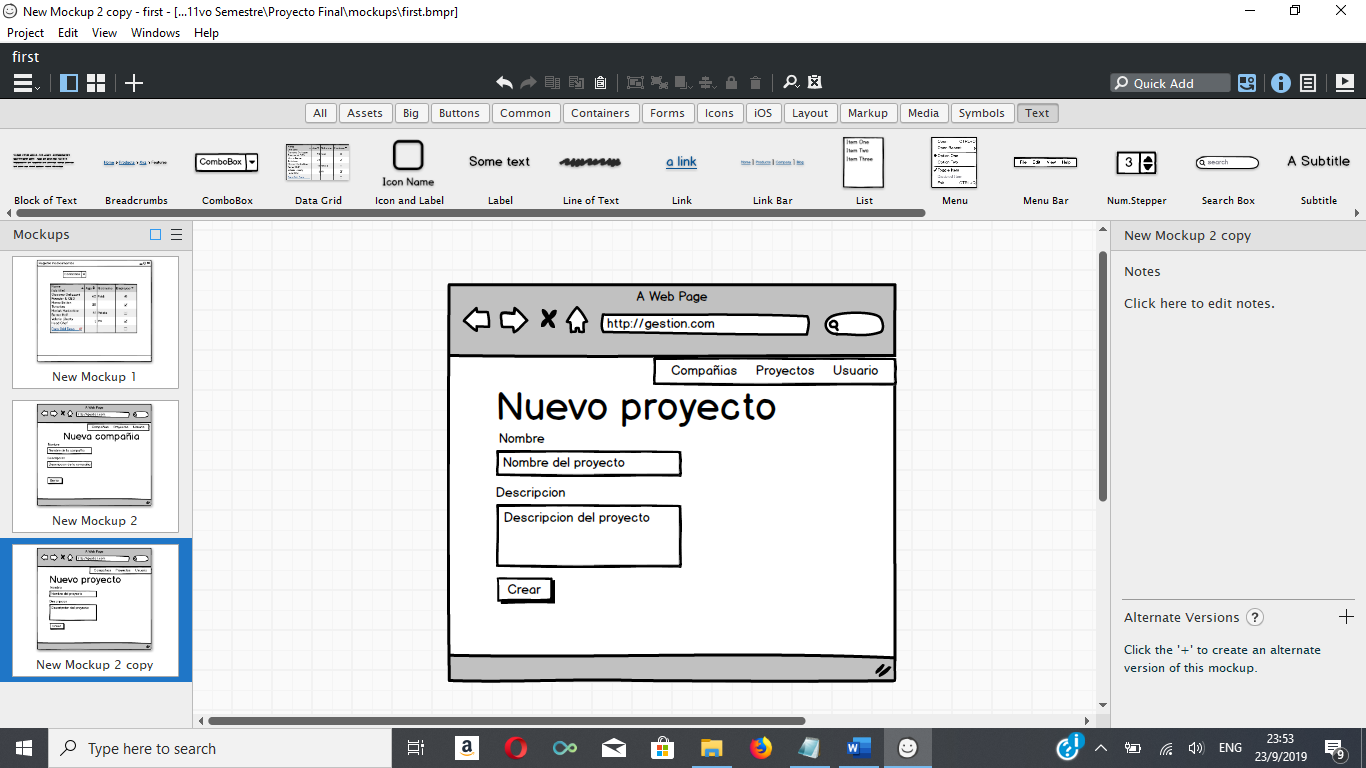
|  |  |  |  |
| --- | --- | --- | --- |
| ID: 5 | Título: Un administrador podrá controlar las cuentas de los usuarios. | | |
| Est. Esf.: | Prioridad: media | Dependiente de: | |
| Descripción:   * El administrador podrá activar o desactivar la cuenta de un usuario. * El administrador tendrá la opción de eliminar las cuentas registradas. | | | |
| Prueba de aceptación:   1. El usuario modificado podrá o no entrar al sistema. 2. El usuario eliminado no podrá ingresar al sitio. | | | |
| Elaborado por:  Ronald Miranda | Responsable:  Ronald Miranda | Fecha de Elaboración:  28/08/2019 | Versión:  1 |

### 3.2.2 Prototipos de entrada y salida

Con el propósito de facilitar el diseño del sistema web se diseñaron los prototipos que ilustran los modelos de interfaces del sistema para el primer sprint.

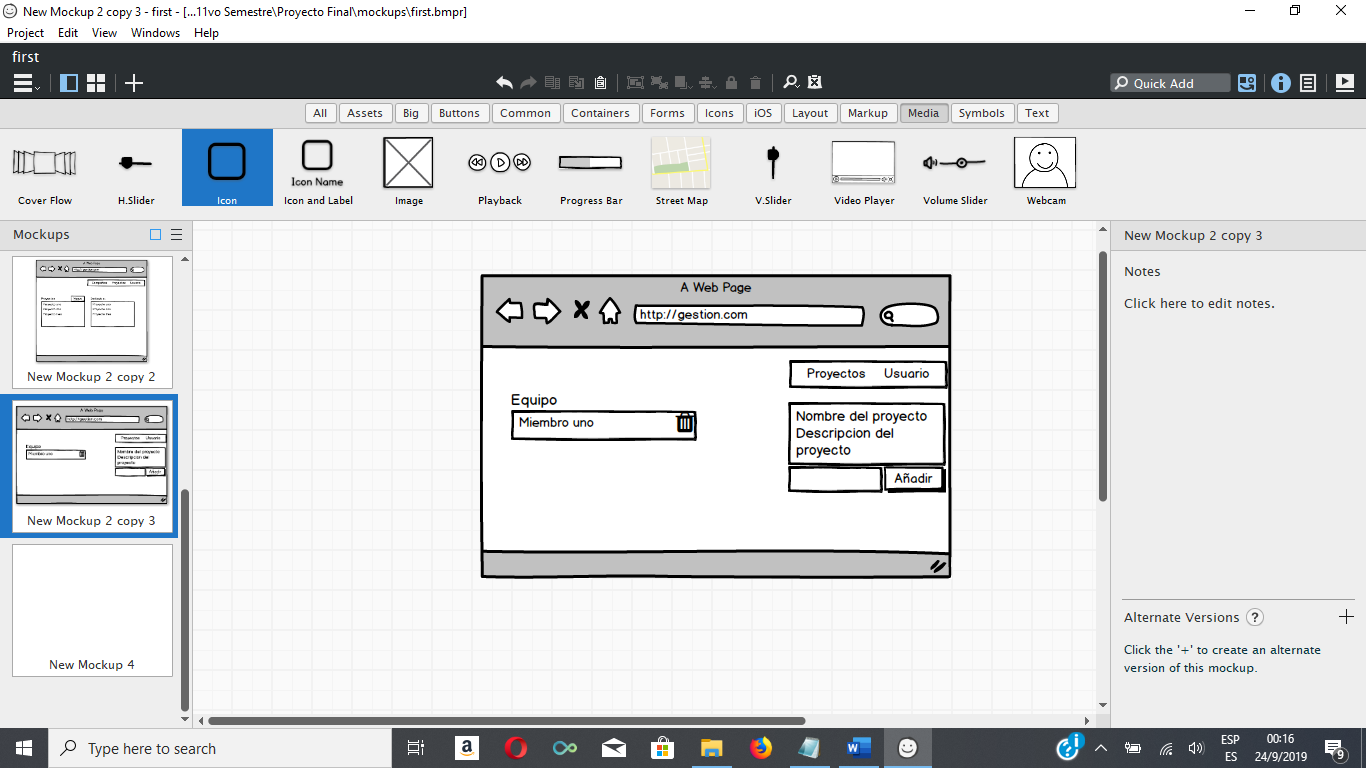
Creación de proyecto

Para este prototipo de entrada se muestra el formulario para la creación de un proyecto, la cual estará disponible para un usuario registrado.



Formulario para añadir usuarios al proyecto

En la siguiente imagen se muestra el formulario para registro de usuarios al proyecto.



### 3.2.3 Modelo de datos

Para esta sección se tienen las tablas y sus respectivos atributos de la base de datos que se identificaron para esta primera etapa. Este modelado contribuyó a la definición de los modelos necesarios en el framework laravel para la elaboración del software.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabla: usuarios | | | | | |
| Descripción: Tabla de usuario registrados. | | | | | |
| Número | Llave | Dato | Descripción | Tipo (longitud) | Observación |
| 1 | \* | id | Número de usuario | I(10) |  |
| 2 |  | nombres | Nombres del usuario | C(15) |  |
| 3 |  | apellidos | Apellidos del usuario | C(15) |  |
| 4 |  | email | Correo electrónico del usuario | C(20) |  |
| 5 |  | contraseña | Contraseña de la cuenta del usuario | C(20) |  |
| 6 |  | ciudad | Ciudad donde trabaja el usuario | C(15) | Dato opcional |

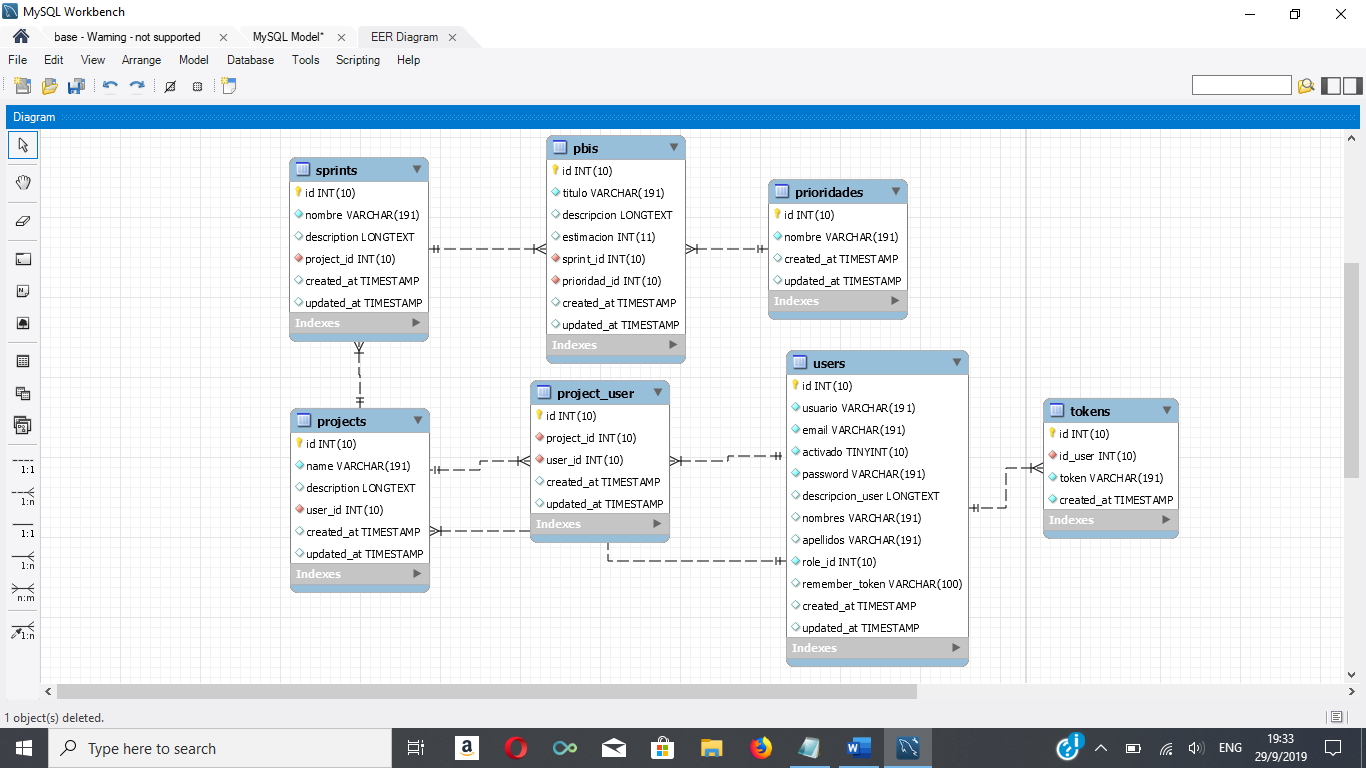
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabla: tokens | | | | | |
| Descripción: Tabla de proyectos registrados. | | | | | |
| Número | Llave | Dato | Descripción | Tipo (longitud) | Observación |
| 1 | \* | id | Número de token | I(10) |  |
| 2 | \* | usuario\_id | Id asociado al usuario | I(10) |  |
| 3 |  | token | token para activar la cuenta del usuario | C(10) |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabla: proyectos\_usuarios | | | | | |
| Descripción: Tabla de proyectos registrados. | | | | | |
| Número | Llave | Dato | Descripción | Tipo (longitud) | Observación |
| 1 | \* | id | Número de tabla pivote | I(10) |  |
| 2 | \* | usuario\_id | Id asociado al usuario | I(10) |  |
| 3 | \* | proyecto\_id | Id asociado al proyecto | I(10) |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabla: proyectos | | | | | |
| Descripción: Tabla de proyectos registrados. | | | | | |
| Número | Llave | Dato | Descripción | Tipo (longitud) | Observación |
| 1 | \* | id | Número de proyecto | I(10) |  |
| 2 | \* | usuario\_id | Id asociado al usuario | I(10) |  |
| 3 |  | descripción | Descripción del proyecto | C(15) |  |
| 4 |  | nombre | Nombre del proyecto | C(15) |  |

### 3.2.4 Diagrama entidad relación de la base de datos

Para este primer diseño de la base de datos, con los datos proporcionados de las tablas anteriores se obtiene el diagrama entidad relación.

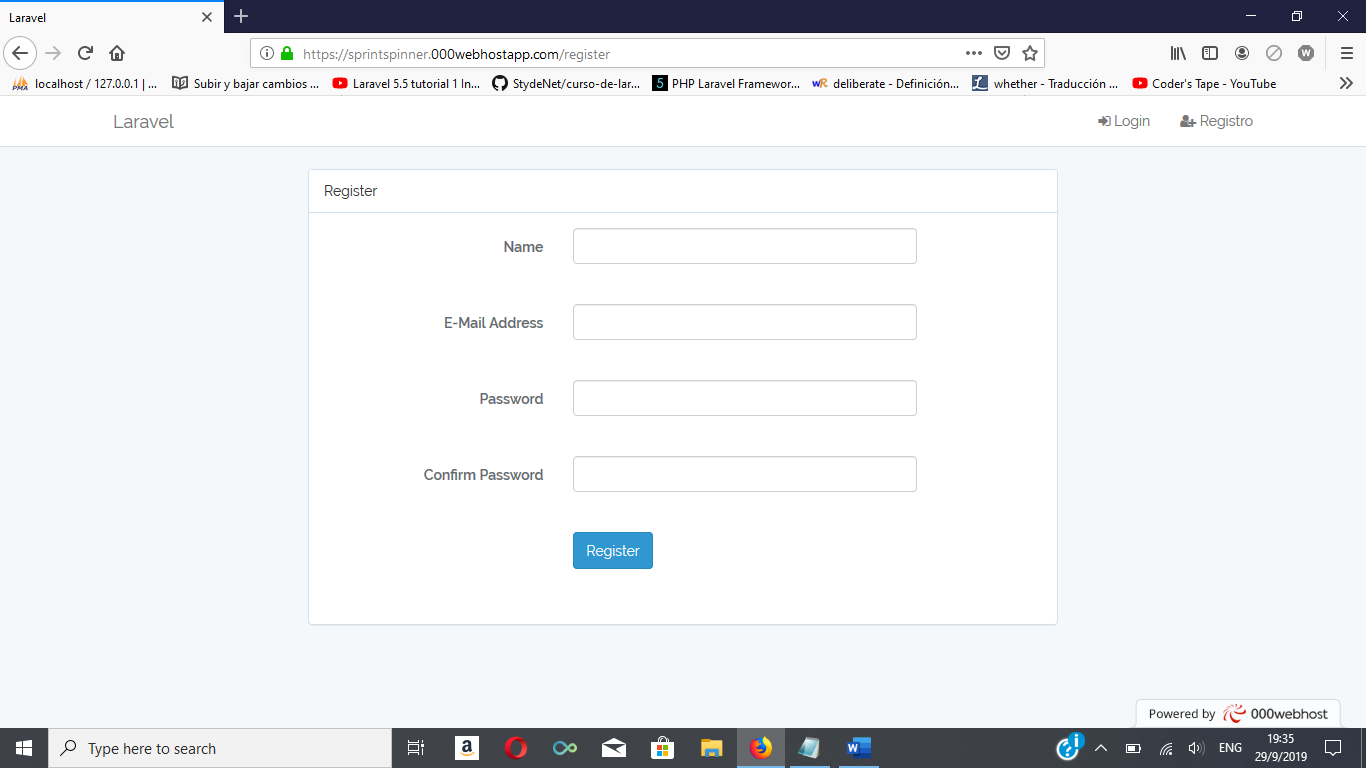


### 3.2.5 Diseño test driven development

Para el diseño de programas se apoyó en la técnica de desarrollo de desarrollo guiado por pruebas, que tiene el objetivo de cumplir los requisitos que se establecieron para el desarrollo.

Registro de usuario

Para este formulario se tienen 4 campos enumerados, como se observa en la siguiente imagen.

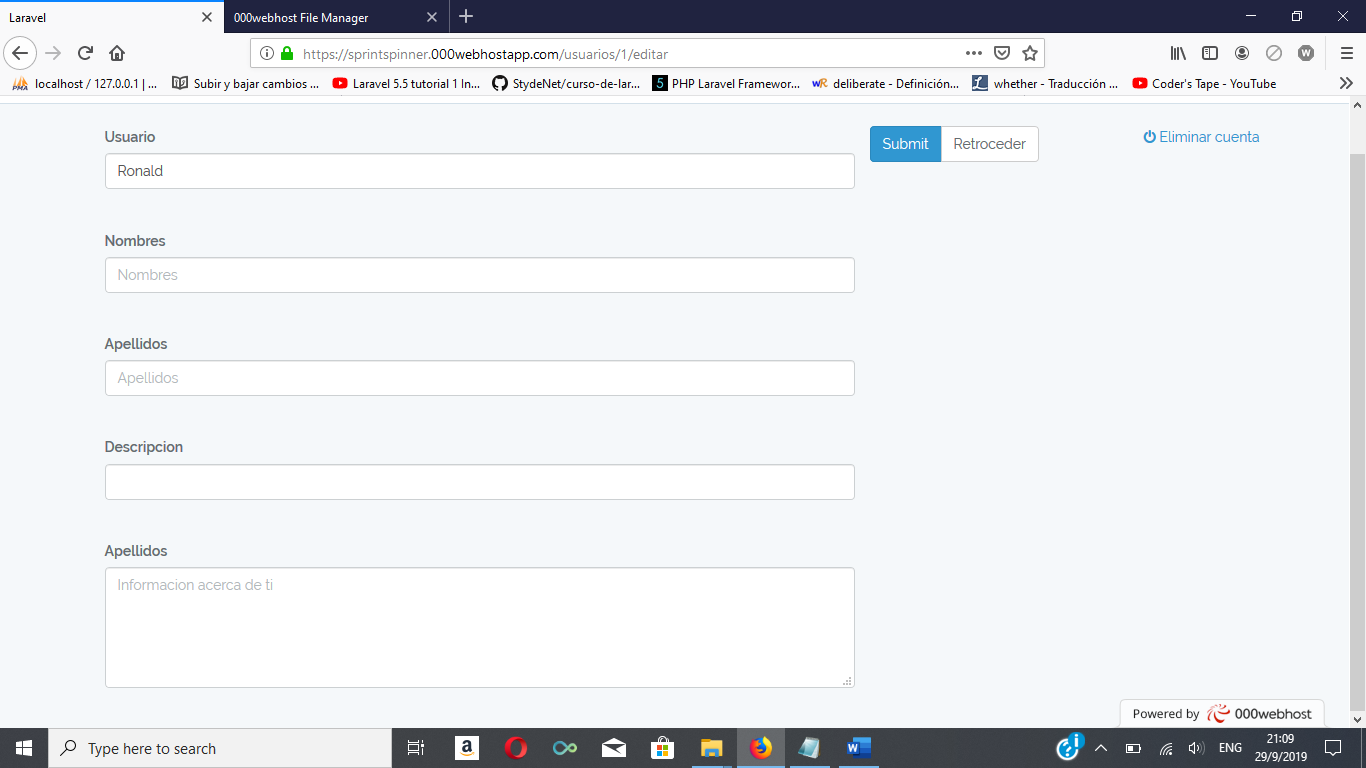


En el siguiente cuadro se muestran los criterios de validación y mensajes respectivos para cada dato.

|  |  |  |  |
| --- | --- | --- | --- |
| Número | Dato | Criterio de validación | Mensaje |
| 1 | usuario | Diferente de espacios | Error en campo nombre |
|  |  | Solo alfabético y numérico | Error en campo nombre |
| 2 | email | Diferente de espacios | Error en campo Email |
|  |  | No debe existir en tabla users | El correo ya existe |
| 3 | password | Diferente de espacios | El campo password está vacío |
|  |  | Mínimo 6 caracteres | El campo password debe tener mínimo 6 caracteres |
| 4 | Confirmar\_password | Diferente de espacios | El campo password está vacío |
|  |  | Mínimo 6 caracteres | El campo password debe tener mínimo 6 caracteres |
|  |  | Deben coincidir password y confirmar\_password | Las contraseñas no son iguales |

Modificación de datos del usuario

Para este formulario se tienen 6 campos enumerados en la siguiente imagen.

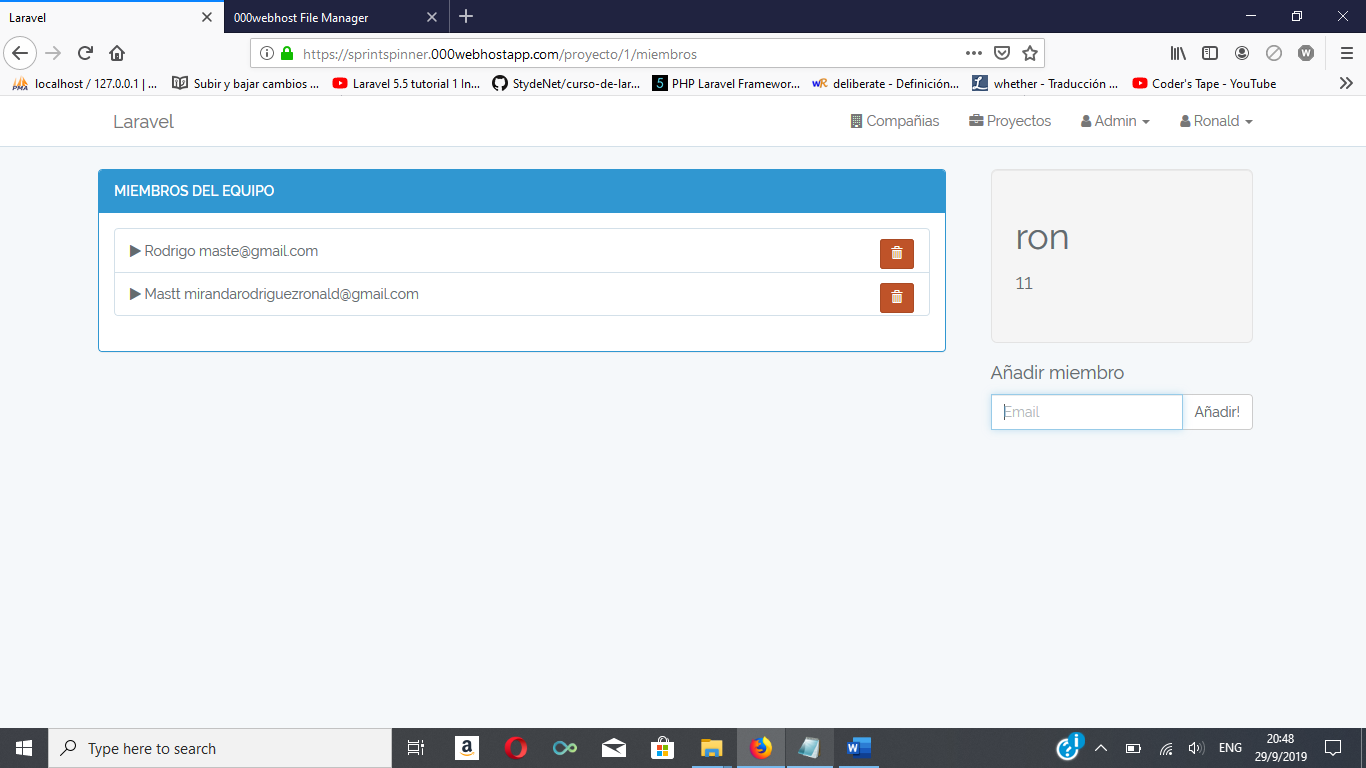


A continuación, se muestran los datos y mensajes que son el resultado de cada criterio de validación para el formulario de modificación.

|  |  |  |  |
| --- | --- | --- | --- |
| Número | Dato | Criterio de validación | Mensaje |
| 1 | usuario | Diferente de espacios | Error en campo nombre |
|  |  | Solo alfabético y numérico | Error en campo nombre |
| 2 | Email | Diferente de espacios | Error en campo Email |
|  |  | No debe existir en tabla users | El correo ya existe |

Formulario de registro de miembros al equipo

En la siguiente imagen se muestra un campo email, para invitar a un usuario del sistema al equipo de trabajo.

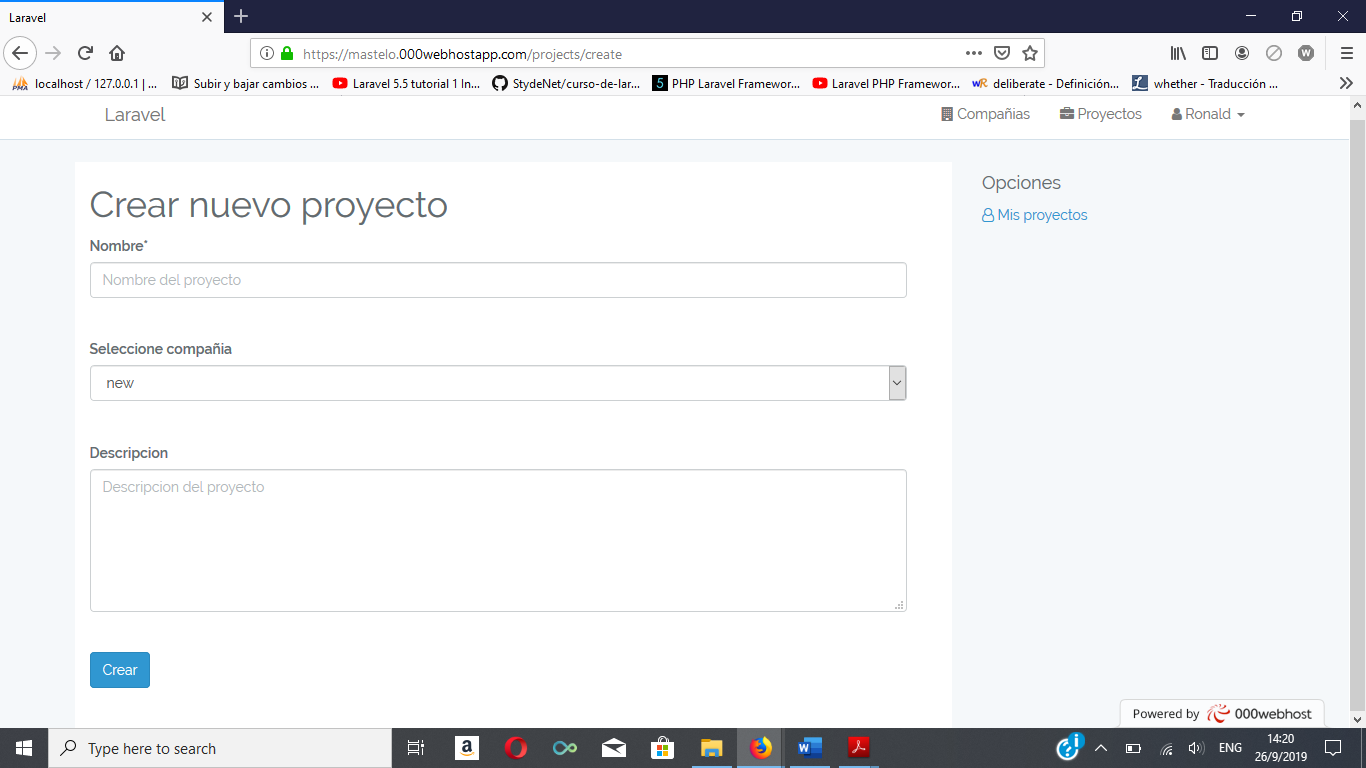


Para las validaciones del campo email, se tiene el siguiente cuadro.

|  |  |  |  |
| --- | --- | --- | --- |
| Pantalla: Pant01 | | | |
| Descripción: Pantalla de registro de miembros al proyecto | | | |
| Número | Dato | Criterio de validación | Mensaje |
| 1 | email | Diferente de espacios | Error en campo email |
|  |  | Tipo email | El campo Email es de tipo correo |
|  |  | No debe existir en tabla project\_user | El usuario ya es miembro |

Formulario para registro de proyectos

En la imagen posterior se muestra el formulario para registrar proyectos en la base de datos.



Los campos probados se muestran con mayor detalle en la próxima tabla.

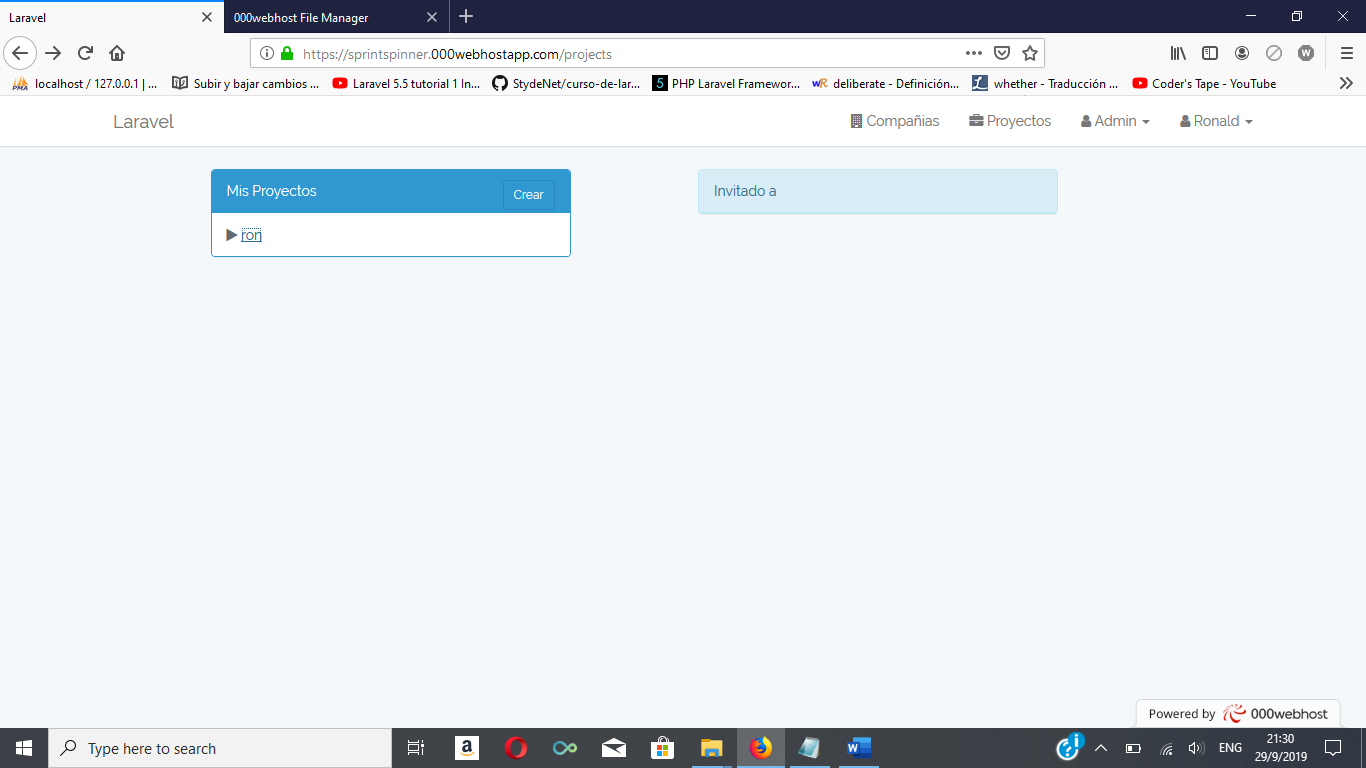
|  |  |  |  |
| --- | --- | --- | --- |
| Número | Dato | Criterio de validación | Mensaje |
| 1 | Nombre | Diferente de espacios | Error en campo nombre |
|  |  | Solo alfabético y numérico | Error en campo nombre |
|  |  | No debe existir en tabla proyectos | El proyecto ya está existe |
| 2 | Descripción | Diferente de espacios | Error en campo descripción |

### 3.2.5 Implementación

Para demostrar el funcionamiento del sistema, se muestra una colección de capturas del sistema web acompañado de una breve descripción para las mismas.

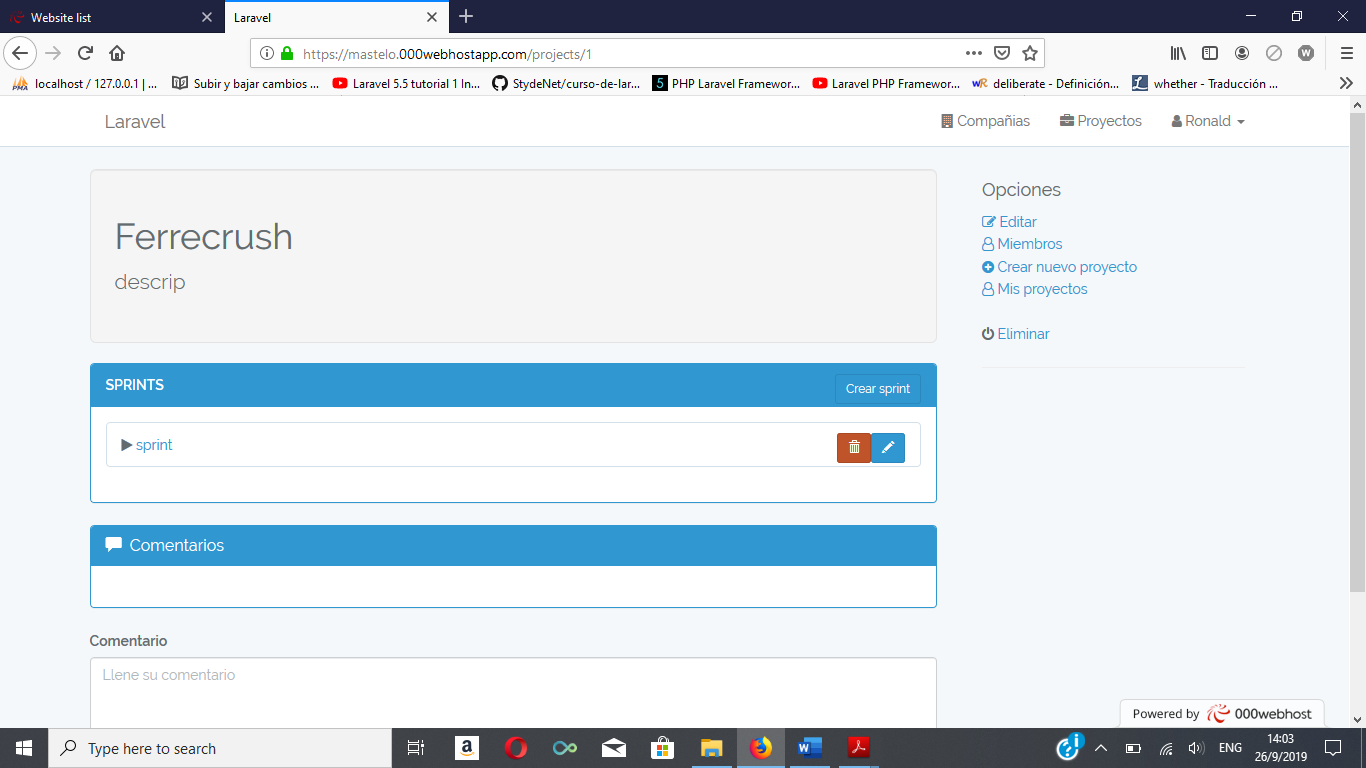
Visualización de los proyectos creados y a los que fue invitado

En la imagen se tienen dos listas para los proyectos que fueron creados por el usuario y otra para los proyectos de los que el usuario es miembro.



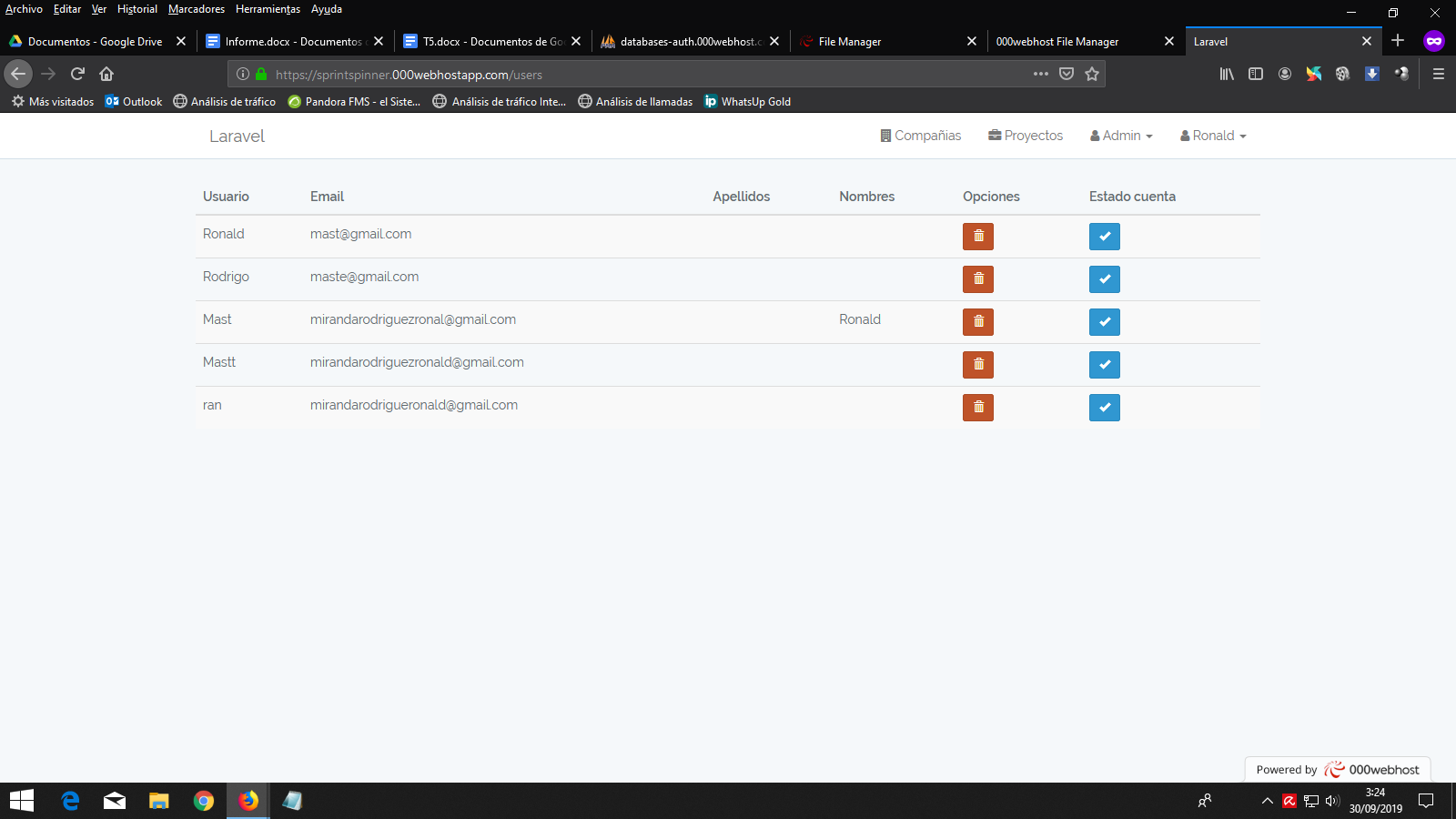
Vista del proyecto

En la imagen se observa los datos y las opciones disponibles para el proyecto.



Control de las cuentas registradas

El administrador podrá eliminar las cuentas registradas, además de cambiar sus estado de activado a desactivado y viceversa. Tambien podra ver los datos relacionados a cada cuenta, como se ve en la posterior imagen.



Código fuente de para añadir usuarios al proyecto

public function anadir(Request $request){

$project = Project::find($request->input('project\_id'));

//if(Auth::user()->id == $project->user\_id){

$user = User::where('email', $request->input('email'))->first();

//revisar si ya es miembro

$miembro = ProjectUser::where('user\_id',$user->id)

->where('project\_id',$project->id)

->first();

//comprobar si es propietaro del proyecto

$propietario = Project::where('user\_id', $user->id)->get()

->where('id',$project->id)

->first();

//si ya es miembro o propietario del proyecto salir

if($miembro || $propietario){

return redirect()->back()->with('success', ['your message,here']);

}

$project = ProjectUser::create([

'project\_id' => $request->input('project\_id'),

'user\_id' => $user->id

]);

return redirect()->back()->with('success', ['your ,here']);

/\* return redirect()->route('projects.show', ['project'=> $project->id])

->with('errors' , 'Error adding user to project');\*/

}

Código fuente para el registro de proyectos

public function create()

    {

        return view('companias.create');

    }

    public function store(Request $request)

    {

        if(Auth::check()){

            $company = Company::create([

                'name' => $request->input('name'),

                'description' => $request->input('description'),

                'user\_id' => Auth::user()->id

            ]);

            if($company){

                return redirect()

->route('companias.show', ['company'=> $company->id])

                ->with('success' , 'Company created successfully');

            }

        }

            return back()->withInput()

->with('errors', 'Error creating new company');

    }

### 3.2.6 Pruebas

Para evaluar el control de errores se prepararon pruebas de caja negra, específicamente la técnica de particiones de equivalencia.

Prueba para invitar a usuarios a un proyecto

|  |  |  |  |
| --- | --- | --- | --- |
| Condiciones de entrada | Tipo | Clases de equivalencia válida | Clases de equivalencia  no válida |
| Email | Conjunto de caracteres | 1: Cadena de caracteres sin tamaño {'a', 'b', 'c',…,'z'} o {'A','B','C',….,'Z'} | 2: Campo en blanco |
|  |  |  |  |

## 3.3 Sprint backlog – Segundo Sprint

La próxima tabla contiene los requerimientos necesarios para la realización del segundo sprint junto con sus tareas.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requerimiento/Tarea | Estado | Estimación de (hrs) esfuerzo |
| 1 | Registro de sprints para un proyecto.   * Crear nuevo sprint. * Modificar datos del sprint. * Eliminar el sprint registrado. | Concluido  Concluido  Concluido | 5  4  4 |
| 2 | Registro de historias de usuario para un sprint.   * Crear nueva historia de usuario. * Eliminar historias de usuario registradas. * Modificar historias de usuario creadas. | Concluido  Concluido  Concluido | 6  4  3 |
| 3 | El sistema web debe estar disponible desde un servidor.   * Implementar el código para que el sistema funcione desde el servidor. * Conectarse con la base de datos del servidor. | Concluido  Concluido | 6  6 |

### 3.3.1 Historia de usuario

Para esta etapa se describen las historias de usuario del sprint backlog con mayor detalle incluyendo su estimación de esfuerzo, prioridad y pruebas de aceptación.

|  |  |  |  |
| --- | --- | --- | --- |
| ID: 1 | Título: Registro de sprints para un proyecto | | |
| Est. Esf.: | Prioridad: media | Dependiente de: | |
| Descripción:   * Para el registro se contará con un formulario con los campos nombre, descripción y fecha. * Los sprints registrados podrán eliminarse. * El usuario podrá modificar los datos del sprint. * Al modificar los datos del sprint se llenarán los campos correspondientes con la información del sprint. | | | |
| Prueba de aceptación:   1. Ingresar datos de sprints. 2. Los usuarios podrán realizar modificaciones a los datos del sprint. 3. No se podrá registrar el mismo nombre de sprint más de una vez. 4. Eliminar sprints registrados. | | | |
| Elaborado por:  Ronald Miranda | Responsable:  Ronald Miranda | Fecha de Elaboración:  28/08/2019 | Versión:  1 |

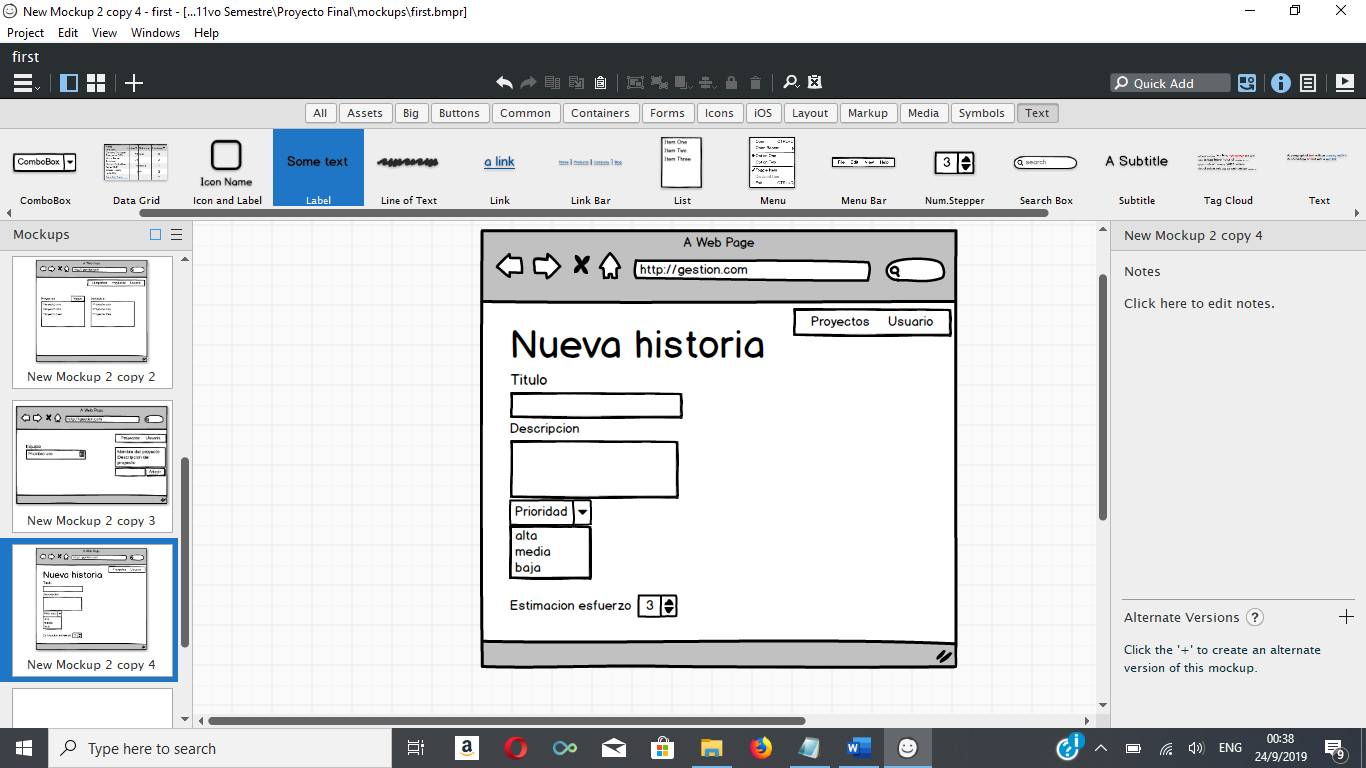
|  |  |  |  |
| --- | --- | --- | --- |
| ID: 2 | Título: Registro de historias de usuario para un sprint | | |
| Est. Esf.: | Prioridad: alta | Dependiente de: | |
| Descripción:   * Para el registro se indicará el nombre, descripción, prioridad y estimación. * Los sprints registrados podrán eliminarse. * El usuario podrá modificar los datos del sprint. * Al modificar los datos del sprint se llenarán los campos correspondientes con la información del sprint. | | | |
| Prueba de aceptación:   1. Registrar nuevas historias de usuario. 2. Los usuarios podrán realizar modificaciones a los datos de la historia de usuario. 3. No se podrá registrar el mismo nombre de historia de usuario más de una vez. 4. Eliminar historias de usuario registradas. | | | |
| Elaborado por:  Ronald Miranda | Responsable:  Ronald Miranda | Fecha de Elaboración:  28/08/2019 | Versión:  1 |

### 

|  |  |  |  |
| --- | --- | --- | --- |
| ID: 3 | Título: El sistema web debe estar disponible desde un servidor. | | |
| Est. Esf.: | Prioridad: alta | Dependiente de: | |
| Descripción:   * Se podrá acceder al sistema web desde un navegador. * Se debe poder interactuar con la base de datos. | | | |
| Prueba de aceptación:   1. Acceder al sitio desde un navegador. 2. Ingresar al sistema con una cuenta registrada. | | | |
| Elaborado por:  Ronald Miranda | Responsable:  Ronald Miranda | Fecha de Elaboración:  28/08/2019 | Versión:  1 |

### 3.3.2 Prototipos de entrada y salida

Para esta fase del diseño de prototipos se muestran los modelos preliminares de las interfaces esperadas.



### 3.3.3 Modelo de datos

En las posteriores tablas se describe la información para cada tabla de la base de datos contemplada para este segundo sprint.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabla: usuarios | | | | | |
| Descripción: Tabla de usuario registrados. | | | | | |
| Número | Llave | Dato | Descripción | Tipo (longitud) | Observación |
| 1 | \* | id | Numero de usuario | I(10) |  |
| 2 |  | nombres | Nombre del usuario | C(15) |  |
| 3 |  | apellidos | Apellido del usuario | C(15) |  |
| 4 |  | email | Correo electrónico del usuario | C(20) |  |
| 5 |  | contraseña | Contraseña de la cuenta del usuario | C(20) |  |
| 6 |  | ciudad | Ciudad donde trabaja el usuario | C(15) | Dato opcional |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabla: proyectos\_usuarios | | | | | |
| Descripción: Tabla de proyectos registrados. | | | | | |
| Número | Llave | Dato | Descripción | Tipo (longitud) | Observación |
| 1 | \* | id | Número de tabla pivote | I(10) |  |
| 2 | \* | Usuario\_id | Id asociado al usuario | I(10) |  |
| 3 | \* | Projecto\_id | Id asociado al proyecto | I(10) |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabla: proyectos | | | | | |
| Descripción: Tabla de proyectos registrados. | | | | | |
| Número | Llave | Dato | Descripción | Tipo (longitud) | Observación |
| 1 | \* | id | Número de proyecto | I(10) |  |
| 2 | \* | usuario\_id | Id asociado al usuario | I(10) |  |
| 3 |  | descripción | Descripción del proyecto | C(15) |  |
| 4 |  | nombres | Nombre del proyecto | C(15) |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabla: sprints | | | | | |
| Descripción: Tabla de los sprint registrados. | | | | | |
| Número | Llave | Dato | Descripción | Tipo (longitud) | Observación |
| 1 | \* | id | Número de sprint | I(10) |  |
| 2 | \* | proyecto\_id | Id asociado al proyecto | I(10) |  |
| 3 |  | descripción | Descripción del sprint | C(25) |  |
| 4 |  |  | del proyecto | C(15) |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabla: pbis | | | | | |
| Descripción: Tabla de historias de usuario registrados. | | | | | |
| Número | Llave | Dato | Descripción | Tipo (longitud) | Observación |
| 1 | \* | id | Número de product backlog item | I (10) |  |
| 2 | \* | sprint\_id | Id asociado al sprint | I (10) |  |
| 3 | \* | prioridad\_id | Id asociado a la prioridad | I (5) |  |
|  |  | titulo | Título del pbi | C (15) |  |
|  |  | descripción | Descripción del pbi | C (70) |  |

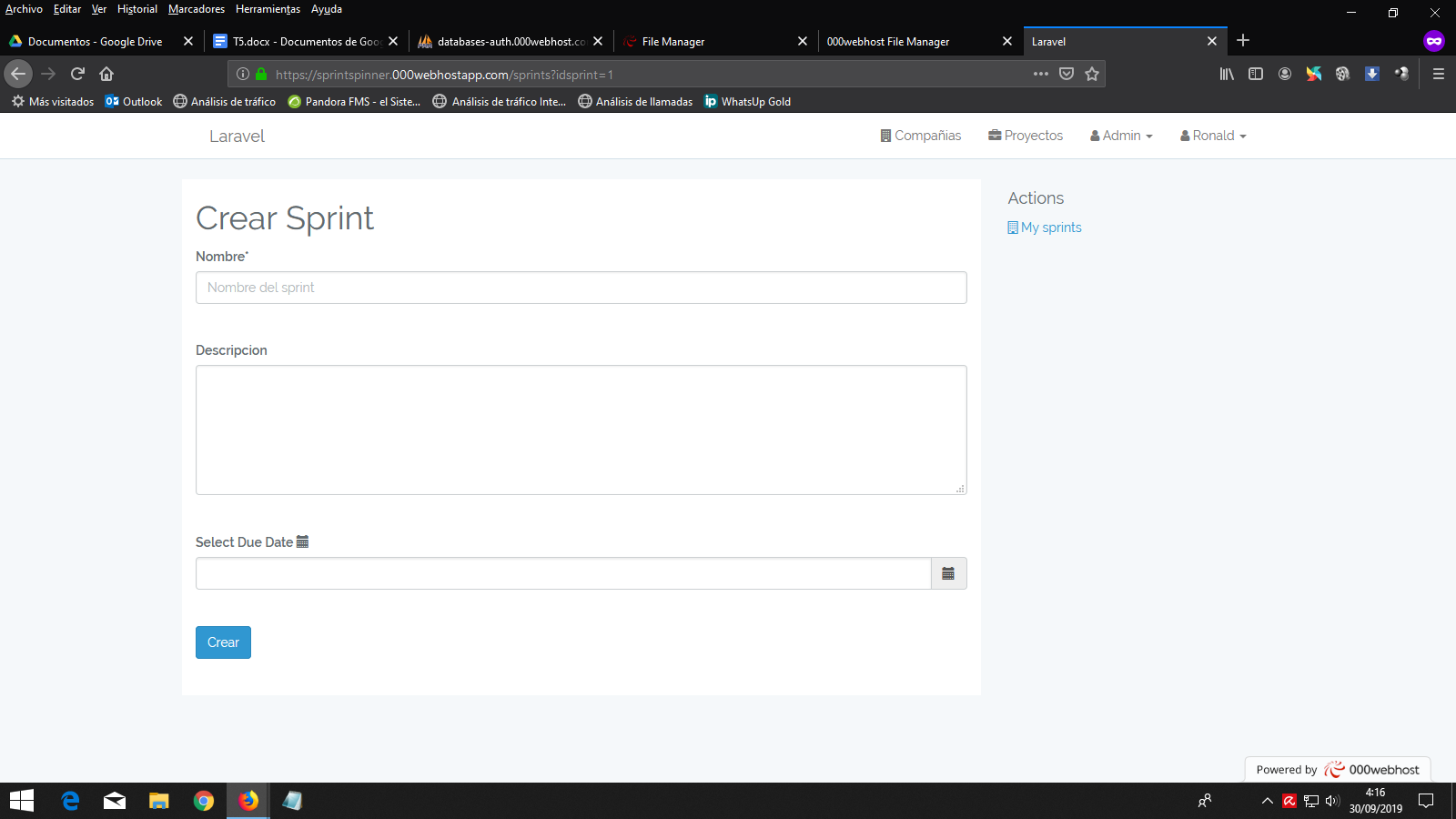
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabla: prioridades | | | | | |
| Descripción: Tabla de historias de prioridad para las historias de usuario. | | | | | |
| Número | Llave | Dato | Descripción | Tipo (longitud) | Observación |
| 1 | \* | id | Numero de prioridad | I (10) |  |
| 2 |  | titulo | Título de la prioridad | C (10) |  |

### 3.3.4 Diseño TDD

En esta fase se preparan las pruebas para validar los datos recibidos,

**Pantalla de registro de sprints**

En el formulario para crear sprints se tienen 3 campos.

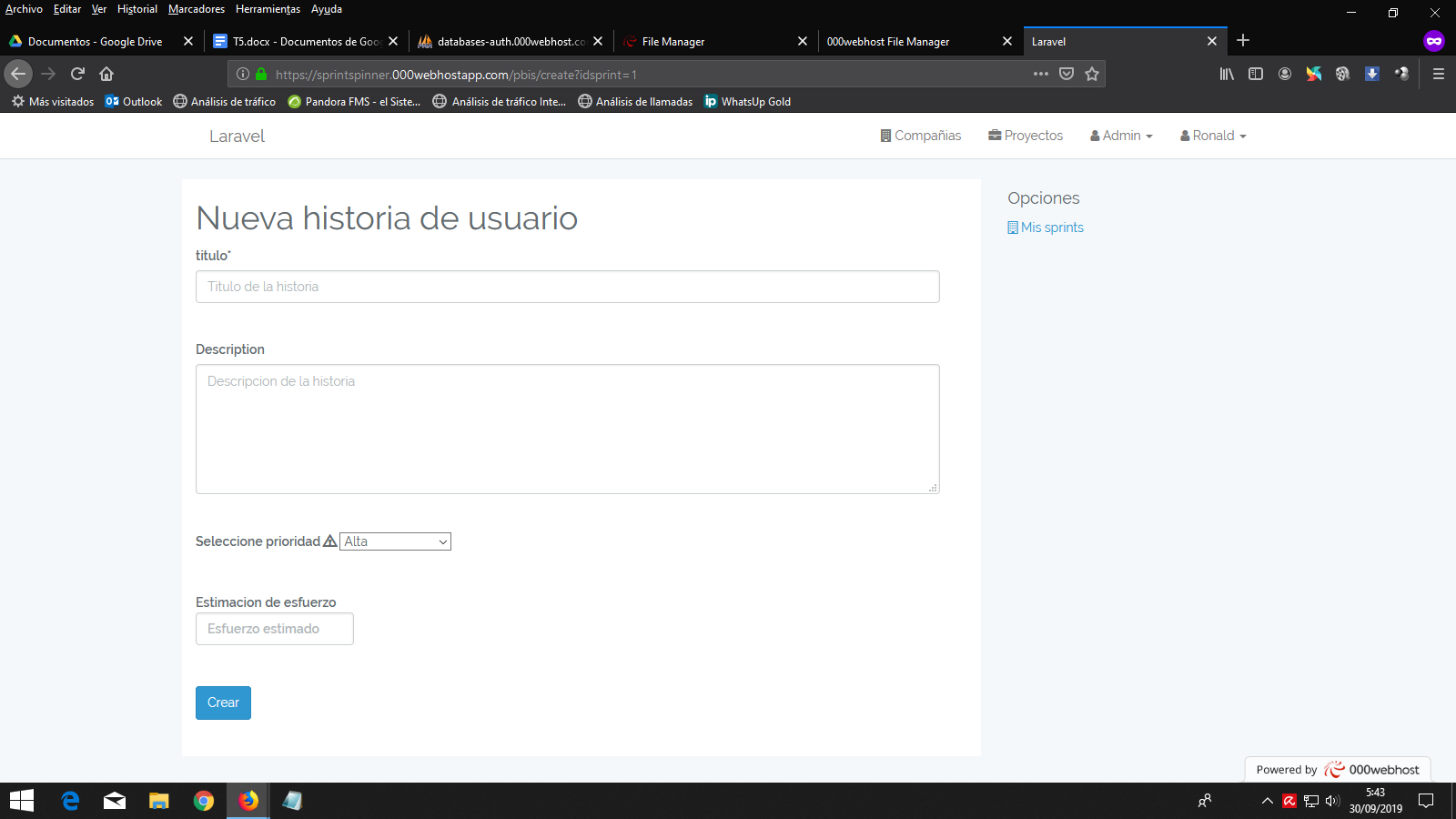


En la siguiente tabla se tienen los datos del formulario, sus criterios de validación con sus respectivos mensajes.

|  |  |  |  |
| --- | --- | --- | --- |
| Número | Dato | Criterio de validación | Mensaje |
| 1 | Nombre | Diferente de espacios | Error en campo nombre |
|  |  | Solo alfabético | Error en campo nombre |
| 2 | Descripción | Diferente de espacios | Error en campo descripción |
|  |  | Solo alfabético | Error en campo descripción |

Formulario de registro de historias de usuario

en la siguiente imagen se tienen los campos



|  |  |  |  |
| --- | --- | --- | --- |
| Pantalla: Pant0 | | | |
| Descripción: Pantalla de registro de pbis | | | |
| Número | Dato | Criterio de validación | Mensaje |
| 1 | Nombre | Diferente de espacios | Error en campo nombre |
|  |  | Solo alfabético | Error en campo nombre |
| 2 | Descripción | Diferente de espacios | Error en campo descripción |
|  |  | Solo alfabético | Error en campo descripción |
|  | Estimación | Solo números | Error en campo estimación |

### 3.3.5 Implementación

### 3.3.6 Pruebas

Sprint backlog – tercer Sprint

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requerimiento/Tarea | Estado | Estimación de (hrs) esfuerzo |
| 1 | Registro de sprints para un proyecto.   * Crear nuevo sprint. * Modificar datos del sprint. * Eliminar el sprint registrado. | Concluido  Concluido  Concluido | 5  4  4 |
| 2 | Registro de historias de usuario para un sprint.   * Crear nueva historia de usuario. * Eliminar historias de usuario registradas. * Modificar historias de usuario creadas. | Concluido  Concluido  Concluido | 6  4  3 |
| 3 | Un usuario puede administrar los documentos de las historias de usuario.   * Eliminar documentos de la h. u. * Añadir documentos a la historia de usuario * Descargar documentos de la historia de usuario | Concluido  Concluido  Concluido | 6  7  6 |

Historias de usuario

Prototipos de entrada y salida

Modelo de datos

Diseño TDD

Implementacion

Pruebas

Sprint backlog – tercer Sprint

Historias de usuario

Prototipos de entrada y salida

Modelo de datos

Diseño TDD

Implementacion

Pruebas

Varias definiciones sistemas de información,

Aumentar sistemas web

No usar proyecto de grado.

Citas, unir capítulos, conectar subtitulos, tratar de hacer otra imagen scrum

Separar toda la teoría en un capítulo.

No hablar de proyecto,

Redactar en futuro.

Rehacer partes scrum y cap 3

# **BIBLIOGRAFIA**

Domínguez, L. (2012). *Análisis de sistemas de información.* Tlalnepantla, México: RED TERCER MILENIO S.C.

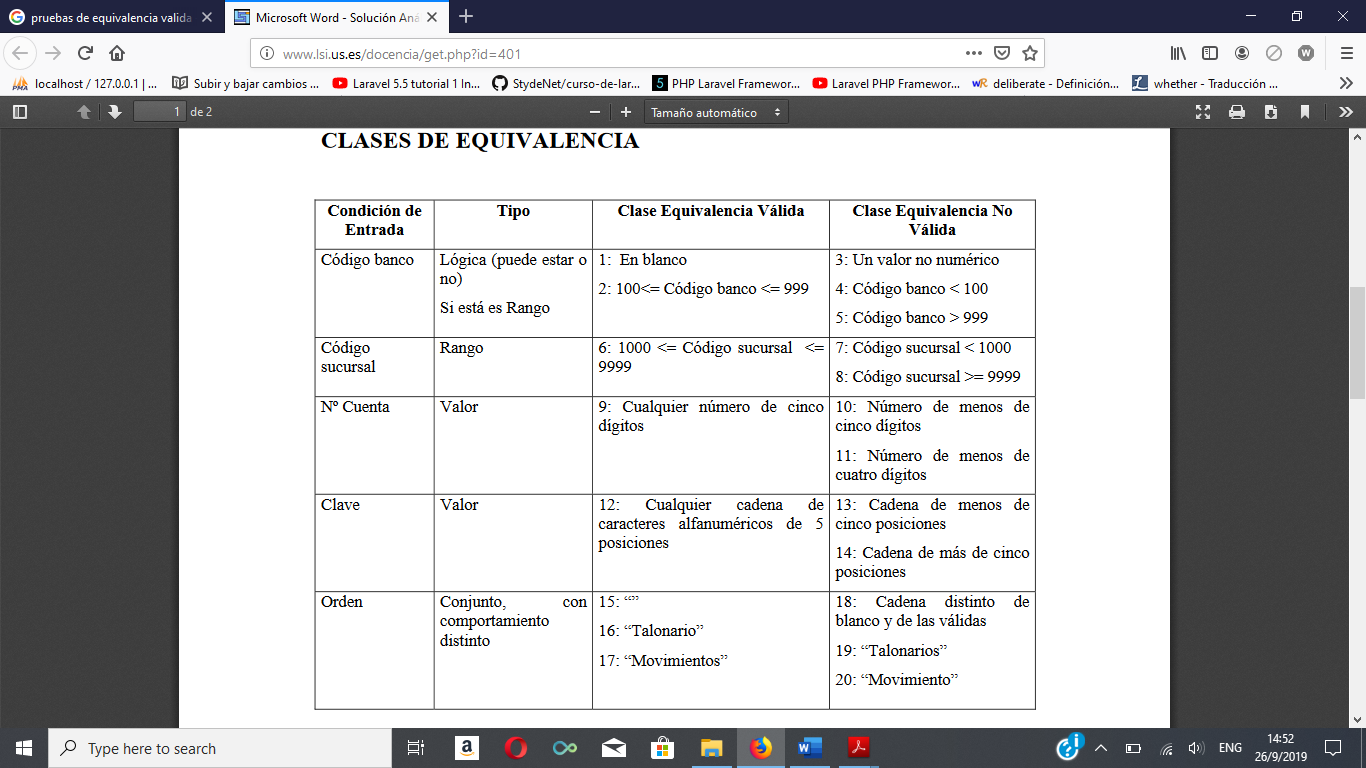
Guerra, Lautaro; Bedini, Alejandro (2005), *Gestión de proyectos de software*, 2ª Edición, Valparaíso, Chile.

Oz, E. (2008). *Administración de los sistemas de información.* México D.F., México: Cengage Learning

Sommerville, I. (2005). *Ingeniería de Software.* Pearson educación

Trello le permite trabajar de manera más colaborativa y ser más práctico. (2019, October 27). Recuperado de https://trello.com/home

|  |  |  |  |
| --- | --- | --- | --- |
| Condiciones de entrada | Tipo | Clases de equivalencia válida | Clases de equivalencia  no válida |
| Carrera | Campo de selección | 1: Licenciatura en ingeniería de Sistemas | 3: Campo en blanco |
|  |  | 2: Licenciatura en ingeniería de Informática |  |
| Título del perfil | Conjunto de caracteres | 4: Cadena de caracteres sin tamaño {'a', 'b', 'c',…,'z'} o {'A','B','C',….,'Z'} o espacios en blanco | 5: Símbolos, números |
|  |  |  | 6: Campo sin llenar |
| Descripción del perfil | Conjunto de caracteres | Cadena de caracteres sin tamaño {'a', 'b', 'c',…,'z'} o {'A','B','C',….,'Z'} o espacios en blanco | Símbolos, números |
|  |  |  | Campo sin llenar |
| Formulario de aprobación | Adjuntar archivo | Formato pdf | cualquier otro formato |
|  |  | nombre del archivo cadena de caracteres sin tamaño {'a', 'b', 'c',…,'z'} o {'A','B','C',….,'Z'} o | Símbolos, números , acentos, ñ |
|  |  |  | sin seleccionar archivo |
| Modalidad de titulación (cmb) | Campo de selección | Proyecto de grado | Campo en blanco |
|  |  | Otra opción |  |
| Área del perfil (cmb) | Conjunto de valores | Cadena de caracteres sin tamaño {'a', 'b', 'c',…,'z'} o {'A','B','C',….,'Z'} o espacios en blanco | Símbolos, números |
|  |  |  | Campo sin llenar |
| Sub-área del perfil (Opcional) | Conjunto de caracteres | Cadena de caracteres sin tamaño {'a', 'b', 'c',…,'z'} o {'A','B','C',….,'Z'} o espacios en blanco | Símbolos, números |
|  |  | Campo dejado en blanco |  |
| Segunda área (Opcional) | Conjunto de caracteres | Cadena de caracteres sin tamaño {'a', 'b', 'c',…,'z'} o {'A','B','C',….,'Z'} o espacios en blanco | Símbolos, números |
|  |  | Campo dejado en blanco |  |
| Tutor | Campo de selección | Escoger un tutor | Campo en blanco |
| Segundo tutor | Campo de selección | Escoger un tutor | Campo en blanco |
|  |  | dejar opción por defecto |  |
| Docente de la materia | Campo de selección | Escoger un docente | dejar opción por defecto |



Mejorar historias eliminar tareas.

Definiciones

Crear de nuevo capítulo 3 ¿?

Con todo lo referente a gestión de proyectos de software

Proceso SCRUM!!! Capitulo 3 de ser necesario

Captura de requerimientos

Historias de usuario

Tareas