

PROTOTIPO DE SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE
PROYECTOS SOFTWARE BAJO LA METODOLOGÍA SCRUM

HEINER WBEIMAR ANGARITA MALDONADO
LUIS GABRIEL TOSCANO RODRIGUEZ

UNIVERSIDAD INDUSTRIAL DE SANTANADER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2013

PROTOTIPO DE SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE
PROYECTOS SOFTWARE BAJO LA METODOLOGÍA SCRUM

HEINER WBEIMAR ANGARITA MALDONADO
LUIS GABRIEL TOSCANO RODRIGUEZ

TESIS DE GRADO: TRABAJO DE INVESTIGACIÓN PARA OPTAR AL TITULO DE
INGENIERO DE SISTEMAS

DIRECTOR: SERGIO FERNANDO CASTILLO CASTELBLANCO
DOCENTE EGRESADO PLANTA

UNIVERSIDAD INDUSTRIAL DE SANTANADER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2013

AGRADECIMIENTOS

Queremos brindar nuestro más sincero agradecimiento a los miembros de la Escuela de Sistemas e Informática que han sido una guía constante a lo largo de este camino; a nuestras familias que llenaron con su compañía y motivación nuestras vidas y comparten hoy nuestros logros y nuestra alegría. Un agradecimiento más específico a nuestro director de proyecto por su excelente labor (pues no hemos sido un grupo fácil) y a Daniel Gualdrón por aportarnos la experiencia que necesitábamos para dar los últimos pasos.

TABLA DE CONTENIDO

INTRODUCCIÓN	12
1. Descripción del proyecto	13
1.1 Descripción de la problemática.....	13
1.2 OBJETIVO	14
1.2.1 GENERAL	14
1.2.2 ESPECÍFICOS	14
1.3 Alcances.....	15
2. Marco teórico y estado del arte	17
2.1 Fundamentos teóricos (SCRUM Development Process)	17
2.1.1 Qué es SCRUM?	17
2.1.2 Qué significa que SCRUM sea una metodología ágil?	18
2.1.3 Historia de SCRUM	19
2.1.4 Roles en SCRUM	20
2.1.5 Eventos en SCRUM	22
2.1.6 Documentos en SCRUM	27

2.2	Tecnologías aplicadas	33
2.2.1	Servidor:	33
2.2.2	Cliente:	36
2.3	ESTADO DEL ARTE	40
2.3.1	OnTime Scrum:	40
2.3.2	Scrumwise:	40
2.3.3	Scrumy.....	40
2.3.4	Cuadro comparativo de funcionalidades:.....	41
3.	Fase preliminar de SCRUM	42
3.1	Historias de Usuario (User Stories)	42
3.2	Product Backlog (Listado del Producto)	43
3.3	Estimación de Entregable y Niveles de Esfuerzo	45
3.4	Duración de Proyecto e Iteraciones	47
3.5	Planeación de Primera Iteración	47
4.	Desarrollo e Implementación	48
4.1	Cambios realizados en el Listado del Producto	48
4.2	Desarrollo de la base de datos	50
4.3	Lista de Pendientes de Iteraciones (Completa)	52

5.	Producto	55
6.	Pruebas.....	78
7.	Conclusiones y Recomendaciones.....	79
7.1	Conclusiones	79
7.2	Recomendaciones	80
8.	Glosario	82
9.	Bibliografía.....	85

RESUMEN

Título: PROTOTIPO DE SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE PROYECTOS SOFTWARE BAJO LA METODOLOGÍA SCRUM

Autor: Heiner Wbeimar Angarita Maldonado / Luis Gabriel Toscano Rodríguez

Palabras clave: SCRUM web sistema información metodología

CONTENIDO

A lo largo de este documento se describe el proceso realizado por los autores en el desarrollo del proyecto. Este proceso se encuentra dividido en diferentes etapas que se ven representadas en cada uno de los capítulos y brevemente descritas en las siguientes secciones:

Marco Teórico y Estado del Arte

Esta sección describe de manera resumida la investigación previa realizada por el Equipo de Trabajo para el desarrollo del proyecto. Se incluyen los aspectos teóricos que explican la metodología SCRUM y sus principales características, eventos y documentos relacionados. De igual forma se describen a profundidad las herramientas utilizadas a lo largo del desarrollo y las razones por las que fueron escogidas por el Equipo de Trabajo. Se incluye también un cuadro comparativo del proyecto frente a herramientas similares existentes en la actualidad.

Fase Preliminar de SCRUM

Esta sección describe los eventos previos a la fase de desarrollo; esto incluye la elaboración de historias de usuario, su transformación a elementos del Listado del Producto y su estimación de niveles de esfuerzo. También se describe en esta sección la estimación temporal de proyecto e iteraciones y el resultado de la Reunión de Planeación de la primera iteración.

Desarrollo e Implementación

Esta sección describe las impresiones del Equipo de Trabajo acerca de las iteraciones realizadas a lo largo de la fase de desarrollo del proyecto; esto incluye la descripción resumida de la metodología utilizada, los cambios ocurridos en el Listado del Producto, el desarrollo y diagrama de la base de datos y El listado final de Pendientes para cada iteración.

Producto

Esta sección presenta el resultado final del proyecto y describe paso a paso el uso de la herramienta mediante ayudas gráficas y explicación detallada.

SUMMARY

Title: PROTOTIPO DE SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE PROYECTOS SOFTWARE BAJO LA METODOLOGÍA SCRUM

Author: Heiner Wbeimar Angarita Maldonado / Luis Gabriel Toscano Rodríguez

Keywords: SCRUM web system information methodology

CONTENT

Throughout this document the process undertaken by the authors in the development of the project is described. This process is divided into several stages that are represented in each of the chapters and briefly described in the following sections:

Theoretical Framework and State of the Art

This section will briefly describe the research conducted by the Team to develop the project. Theoretical aspects are included that explain the SCRUM methodology and its main features, events and related documents. Likewise the tools used during the development and the reasons why they were chosen by the Team are described. A comparative table of the project against similar tools existing today is also included.

SCRUM's Preliminary Phase

This section describes the events leading up to the start of the development phase and the first Sprint. Development of user stories, further transformation to the Product Backlog Items and estimating levels of effort of each one are included. It is also described in this section project and sprint's time estimation and the result of the first Sprint Planning Meeting.

Development and Implementation

This section describes the impressions of the Team on the sprints in the development phase of the project. Description of the methodology used, changes in the Product List, database development and diagram and the final Sprint Backlog are included.

Product

This section presents the project's final result and describes step by step the use of the tool by using visual aids and detailed explanation.

INTRODUCCIÓN

Es para algunas personas aún una creencia común que el proyecto ideal es aquel que debiera funcionar como una máquina recién ajustada; es decir, un proceso lineal y secuencial donde la entrada correcta y un proceso de calidad garantizaran un resultado igualmente satisfactorio para todas las partes involucradas. Esto sin embargo, ha dejado de ser cierto en un mundo comercial donde la flexibilidad y la velocidad se han erigido como estandartes de las buenas prácticas en manejo de proyectos y se ha definido no el proyecto ideal; sino mas bien el realista, como aquel que funciona en similitud a un equipo deportivo que enfrenta como un todo a un entorno hostil y cambiante para obtener el mejor resultado posible.

Hiroataka Takeuchi e Ikujiro Nonaka, dos destacados docentes universitarios de Japón; fueron los primeros que describieron esta situación en la reconocida revista *Harvard Business Review* en 1986, comparando el manejo de proyectos desde un enfoque holístico con un equipo de rugby; que se mueve en conjunto a través del campo y está dispuesto a pasar el balón atrás y adelante y cambiar de dirección cuando sea necesario para alcanzar la meta final. Estas declaraciones tuvieron un impacto significativo en mentalidades a nivel mundial, desencadenando así prácticas novedosas y metodologías de trabajo holísticas y ágiles para suplir las necesidades del mercado cambiante.

SCRUM se impone hoy en día como una de las resultantes más fuertes de este movimiento; especializada en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad, entre los cuales el desarrollo de software ocupa un lugar primordial en la lista. Grandes empresas del área de la tecnología como Microsoft, Google, Amazon, IBM, Adobe o Sony/Erickson utilizan SCRUM diariamente y reconocen los buenos resultados de esta práctica.

Bajo esta premisa, se ha desarrollado una herramienta software con el objetivo de reducir el factor de error humano en la implementación de esta metodología y entregar información organizada y concisa al usuario que permita realizar retroalimentación del proceso realizado y motivar la mejora continua a los equipos de trabajo que las utilicen; para así contribuir al desarrollo de software competitivo y de alta calidad en Colombia.

1. DESCRIPCIÓN DEL PROYECTO

1.1 Descripción de la problemática

Es un hecho que transcurre una época en la que la tecnología se ha vuelto indispensable para la vida diaria, y el software como motor vital de ésta se ha establecido como una de las industrias más fuertes en el mercado mundial. Sin embargo, el tamaño de los sistemas requeridos en la actualidad continúa creciendo y lo hace también la lista de posibilidades que deslumbran a los clientes que buscan mantener sus productos a lo alto de la vanguardia.

Esto implica la necesidad de reunir grandes equipos de programadores y plantea una problemática que involucra la dificultad intrínseca de manejar correctamente estos equipos. Para ofrecer una solución, han sido creadas metodologías especializadas en el manejo de proyectos software, que plantean prácticas para optimizar los resultados del trabajo de equipo alrededor de un proyecto.

La metodología ágil, una de las más utilizadas y recomendadas a nivel mundial y el marco central del desarrollo de este proyecto; enfrenta esta problemática a través de métodos iterativos e incrementales donde los productos evolucionan con la colaboración de grupos multidisciplinarios que llevan su propio control y minimizan riesgos a través de entregables en lapsos cortos. Gran cantidad de empresas hacen uso de esta metodología con resultados tangibles, sin embargo, los métodos siguen dependiendo en gran parte del ser humano, lo que trae como consecuencia que su eficacia varíe según donde y por quien sea implementada.

Tras identificar esta situación, se decidió desarrollar un sistema de información basado en la web, que permita al encargado de cada equipo de desarrollo llevar un control organizado, detallado y accesible del estado de su proyecto, con una interfaz que facilite la actualización rápida e intuitiva del estado diario y el acceso a medidores gráficos de rendimiento que faciliten el proceso de retroalimentación. Se decidió basar esta herramienta en la metodología ágil SCRUM, pues es una de las más utilizadas y documentadas por empresas de software a nivel mundial y dirige su enfoque a la creación de equipos motivados, capaces de organizarse por sí mismos, de alta comunicación y transparencia total mientras da al usuario y al cliente el más alto protagonismo.

1.2 OBJETIVO

El siguiente listado enuncia de manera organizada y explícita los propósitos del proyecto que orientaron los esfuerzos del equipo de desarrollo a lo largo del marco de trabajo:

1.2.1 GENERAL

Desarrollar un prototipo de sistema de información web para la gestión de proyectos software bajo la metodología SCRUM.

1.2.2 ESPECÍFICOS

- Definir y priorizar las historias de usuario y sus respectivos productos entregables.
- Diseñar y definir las iteraciones (*sprints*).
- Programar el componente software que cumpla los requerimientos detallados en la historia de usuario “Fase Inicial del Proyecto”.
- Programar el componente software que cumpla los requerimientos detallados en la historia de usuario “Fase Preliminar SCRUM”.
- Programar el componente software que cumpla los requerimientos detallados en la historia de usuario “Planeación de iteración”.
- Programar el componente software que cumpla los requerimientos detallados en la historia de usuario “SCRUM diario”.
- Programar el componente software que cumpla los requerimientos detallados en la historia de usuario “Retroalimentación del proyecto”.
- Realizar Pruebas de integración.

1.3 Alcances

Las herramientas actuales para manejo de proyectos SCRUM proporcionan funcionalidades como actualización constante en la nube y sincronización de datos en tiempo real para múltiples usuarios que buscan acercar al Equipo Scrum en un nivel virtual; el desarrollo de esta herramienta estará enfocado principalmente en las tareas del Scrum Master, por lo que estas funciones no serán implementadas.

Algunas aplicaciones de este tipo también ofrecen funcionalidades de acceso a hardware (como activación de alarmas visuales o sonoras del dispositivo), nuestra aplicación será programada totalmente en la web, lo que implica que no tendrá ninguna de estas posibilidades. Ninguno de los elementos mejorados de interfaz de usuario (autocompletado, "arrastrar y soltar", elementos táctiles, etc...) serán implementados ya que no son indispensables para el desarrollo de un prototipo.

Con respecto a la metodología SCRUM, el software se limita en ciertos puntos para evitar una documentación exhaustiva donde solo se pide lo necesario para funcionar correctamente. Se evita la inserción de las historias de usuario y se registran al sistema solo desde el momento en que las se transforman en *product backlog item* (Elemento de la lista del producto).

Al ser una metodología donde el éxito depende en gran parte de las prácticas humanas, la herramienta no interfiere con las interacciones del equipo y solo se da énfasis en responder a las necesidades de documentación del SCRUM Master posterior a cada evento de la metodología; por este motivo, ningún otro rol de SCRUM ha sido incluido en el sistema. Acciones como el registro de cambios de cada daily scrum (el cual es una función del Equipo de Desarrollo) también será llevado a cabo por el Scrum Master por motivos de llevar un control actualizado del estado de su proyecto. De esta manera, los estandartes de SCRUM, el diálogo y la transparencia en sus eventos no se ven afectados.

SCRUM enuncia que el éxito o fallo de una iteración solo puede ser asumido como grupo sin faltas o méritos individuales basándose en los incrementos resultantes y la métrica inamovible de una calidad nunca negociable. Por este motivo, la asignación de un miembro del equipo como encargado de un PBI (Product backlog ítem) o más aún de una tarea, no aporta relevancia y fue eliminado de las funciones del software.

Además, aunque la metodología permite la cancelación de iteraciones en caso de eventos que imposibiliten o cuestionen la relevancia del cumplimiento de la meta, también enuncia que el

equipo de trabajo debe estar preparado para este tipo de imprevistos; la probabilidad de una situación que realmente obligue a este tipo de acciones es realmente baja, razón por la que se decidió que una primera versión del software no debería agregar esta característica.

Por otra parte, debido a las diferencias de calendarios laborales existentes para distintas empresas a nivel nacional, se decidió no limitar el software según las fechas reales de uso del sistema, y llevar el control de tiempos del proyecto a través de la cantidad de *daily scrums* registrados en él. De esta manera se busca proteger tanto la diversidad de usuarios del software como el manejo intuitivo de la herramienta.

2. MARCO TEÓRICO Y ESTADO DEL ARTE

La investigación previa que realiza el equipo de trabajo necesaria para el desarrollo del proyecto tiene como objetivo no solo entender a la perfección las metodologías y tecnologías a aplicarse dentro de éste, sino analizar las herramientas existentes que buscan suplir necesidades similares a las propuestas por el proyecto en cuestión; esto con el objetivo de realizar propuestas innovadoras y relevantes en el área a tratar. Los resultados de la investigación realizada en el marco preliminar de este proyecto se encuentran descritas en las siguientes secciones.

2.1 Fundamentos teóricos (Schwaber)

El éxito del desarrollo de una herramienta software depende primordialmente del correcto entendimiento del proceso a optimizar. El software desarrollado a lo largo de este proyecto comprende la optimización de los eventos y documentación consignados dentro de la metodología SCRUM y el fundamento teórico descrito en las siguientes secciones comprende las características mas importantes de esta metodología.

2.1.1 Qué es SCRUM?

SCRUM es una metodología iterativa e incremental de gestión, mejora y mantenimiento de sistemas existentes o prototipos en producción. Sus resultados incluyen principalmente software orientado a objetos y están basados principalmente en las siguientes variables:

1. Requerimientos del cliente: ¿Cómo debe avanzar el sistema?
2. Presión del tiempo: ¿Qué lapso de tiempo es necesario para ganar una ventaja competitiva?
3. Competencia: ¿Qué ofrece la competencia y cómo puede superarse?
4. Calidad: ¿Cuál es la calidad permitida según los 3 anteriores?
5. Visión: ¿Qué cambios se requieren para alcanzar la visión del sistema?
6. Recursos: ¿Cuánta gente y dinero hay disponibles? ¹

SCRUM fue implementado como respuesta contra las prácticas de programación demasiado estructuradas y estrictas, que convertían el proceso de desarrollo en algo burocrático, lento, degradante e inconsistente con las expectativas de eficiencia de desarrolladores y clientes.

1 Schwaber, K. Advanced Development Methods. SCRUM Development Process. P 57.

SCRUM se encuentra catalogado como una metodología ágil, un concepto no muy fácil de definir para quien no se encuentra familiarizado con el tema. En la siguiente sección se explica a un nivel básico las características que engloba este término.

2.1.2 Qué significa que SCRUM sea una metodología ágil?

No hay una forma sencilla para definir la agilidad de una metodología, Alistair Cockburn, uno de los iniciadores del movimiento ágil describió esta situación en su libro de la siguiente manera:

“Ágil implica ser efectivo y manejable. Un proceso ágil es ligero pero suficiente al mismo tiempo. Debe ser ligero para permanecer siendo manejable y suficiente para mantenerse dentro del negocio. La pregunta sobre el uso de metodologías ágiles no es si una metodología ágil puede ser usada en una determinada situación, sino ¿cómo podemos nosotros ser y permanecer ágiles en esa situación?”²

Para el desarrollo de este proyecto, se considera ágil al método de ingeniería del software que se base en desarrollo iterativo e incremental con entregables en lapsos cortos llamados iteraciones. En un principio, un método iterativo no parece muy eficiente, pero tal como Gary Chin explica en su libro: “Ágil no significa rápido, la velocidad solo es un multiplicador del desarrollo ágil”³ y por eso cada iteración debe realizarse e incluir procesos de: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación aunque estos nunca lleguen al cliente. El verdadero objetivo no es que las cosas se hagan más rápido, sino controlar riesgos, aprender de los errores y adaptarse a los cambios sobre la marcha.

Los valores consignados en el manifiesto ágil, el cual reúne los principios resultantes de la reunión de expertos en métodos alternativos convocados por Kent Beck, creador de la metodología Xtreme Programming (XP) en el año 2001 son los siguientes:

“A través de este trabajo hemos aprendido a valorar:

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación exhaustiva.

² Cockburn, Alistair. *Agile Software Development. Highsmith Series. 2000. p 149.*

³ Chin, Gary. *Agile Project Management: How to Succeed in the Face of Changing Project Requirements. AMACOM. 2004. p 9.*

- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan".⁴

Existe una gran variedad de metodologías que pueden considerarse ágiles. Se decidió SCRUM como marco de referencia para la realización de este proyecto por las características descritas en las siguientes secciones.

2.1.3 Historia de SCRUM

El concepto de SCRUM tiene su origen en un estudio de 1986 escrito por Hirotaka Takeuchi e Ikujiro Nonaka en el Harvard Business Review sobre los nuevos procesos de desarrollo utilizados en productos exitosos en Japón y los Estados Unidos. En este estudio se comparaba la forma de trabajo de estos equipos altamente productivos y multidisciplinares con la colaboración entre los jugadores de Rugby de la siguiente manera:

“En el acelerado y agresivo mundo del desarrollo de nuevos productos, la velocidad y flexibilidad son esenciales. Cada vez más compañías están descubriendo que la antigua forma secuencial de creación de productos sencillamente ya no es suficiente. Empresas en Japón y Estados Unidos están desarrollando un método holístico, similar a un juego de rugby, donde la bola es pasada por cada miembro del equipo, mientras este se mueve en su totalidad a través del campo.

Este enfoque holístico tiene 6 características: inestabilidad incorporada, equipos auto organizados, superposición de fases de desarrollo, multiaprendizaje, control sutil y transferencia organizacional de aprendizaje. Estas seis piezas encajan como un rompecabezas, formando un proceso rápido y flexible para desarrollo de nuevos productos”.⁵

En la década de 1990, Ken Schwaber utilizó un primer intento de SCRUM enfocado al desarrollo de software en su compañía *Métodos Avanzados de Desarrollo*, y Jeff Sutherland, con John Scumniotales y Jeff McKenna, desarrollaron un enfoque similar en *Easel Corporation* y fueron los primeros en referirse a él con la sola palabra SCRUM.⁶

4 *Manifesto for Agile Software Development.*

⁵ *Takeuchi and Nonaka: The New New Product Development Game (Harvard Business Review, Jan-Feb 1986).*

⁶ *Sutherland, Jeff (2004-10). "Agile Development: Lessons learned from the first Scrum"*

En 1995, Sutherland y Schwaber presentaron conjuntamente un documento que describía la metodología SCRUM en el *Taller de Implementación y Diseño de Objetos de Negocio* celebrado como parte del *Object-Oriented Programming, Systems, Languages & Applications '95* (OOPSLA 95) en Austin, Texas. Schwaber y Sutherland colaboraron durante los años siguientes para combinar sus escritos, experiencias y mejores prácticas de la industria para formar lo que ahora se conoce como SCRUM.

Desde 1995 miles de proyectos en todo el mundo han utilizado SCRUM para el desarrollo de productos, tanto en empresas pequeñas como en grandes multinacionales. La SCRUM Alliance es la organización sin ánimo de lucro que se encarga de difundir SCRUM en este ámbito.

La metodología comprende un listado de roles, documentos y eventos que deben realizarse para ser aplicada correctamente, los cuales se encuentran definidos completamente en la guía de SCRUM⁷ de sus creadores Ken Schwaber y Jeff Sutherland, que representa la más completa y confiable referencia de SCRUM abierta al público de manera gratuita. La información consignada en la guía se encuentra resumida en las siguientes secciones.

2.1.4 Roles en SCRUM

Los Equipos SCRUM son auto-organizados y multifuncionales, esto quiere decir que eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo, pero tienen a su vez todas las competencias necesarias para llevar a cabo el trabajo. El modelo de equipo en SCRUM está diseñado para optimizar la flexibilidad, la creatividad y la productividad y está compuesto por los siguientes roles:

Product Owner

Representa la voz del cliente y aporta la visión de negocio, es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo, aunque el cómo se lleva a cabo esto podría variar ampliamente. El *product owner* es la única persona responsable de gestionar el Listado del Producto. Sus funciones incluyen:

- Expresar claramente los elementos del Listado del Producto.

⁷ Sutherland, Jeff/Ken Schwaber; *The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game* (2011-10).

- Ordenar los elementos del Listado para alcanzar los objetivos y misiones de la mejor manera posible.
- Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo.
- Asegurar que el Listado del Producto es visible, transparente y claro para todos, y que muestra aquello en lo que el equipo trabajará a continuación.
- Asegurar que el Equipo de Desarrollo entiende los elementos del Listado al nivel necesario.

El *product owner* podría hacer el trabajo anterior, o delegarlo en el Equipo de Desarrollo, sin embargo, en ambos casos sigue siendo el responsable de dicho trabajo. El *product owner* es una única persona, no un comité, aunque podría representar los deseos de un comité en el Listado del Producto; pero aquellos que quieran cambiar la prioridad de un elemento de la Lista deben hacerlo a través de él.

Para que el *product owner* pueda hacer bien su trabajo, toda la organización debe respetar sus decisiones. Estas decisiones se reflejan en el contenido y en la priorización del Listado del Producto. No está permitido que nadie pida al Equipo de Desarrollo que trabaje con base en un conjunto diferente de requerimientos, y el Equipo de Desarrollo no debe actuar con base en lo que diga ninguna otra persona.

Scrum Master

El Scrum Master es la persona responsable de asegurar que la metodología SCRUM es entendida y adoptada, asegurándose de que el Equipo SCRUM trabaja ajustándose a la teoría, prácticas y reglas de SCRUM. El Scrum Master es un líder que está al servicio del Equipo SCRUM y ayuda además a las personas externas a entender cuáles interacciones con éste pueden ser de ayuda y cuáles no. El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo SCRUM y es quien se encarga de mantener el camino libre de obstáculos para que se pueda alcanzar la meta del proyecto.

Equipo de Desarrollo

El Equipo de Desarrollo es el grupo de profesionales que desempeña el trabajo de entregar un Incremento de producto al final de cada iteración. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento; son estructurados y empoderados para organizar y gestionar su propio trabajo, lo que optimiza su eficiencia y efectividad.

Los Equipos de Desarrollo deben tener las siguientes características:

- Son auto-organizados. Nadie indica al Equipo de Desarrollo cómo convertir elementos del Listado del Producto en Incrementos de funcionalidad potencialmente desplegados.
- Son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un Incremento de producto.
- SCRUM no reconoce títulos para los miembros de un Equipo de Desarrollo, todos son Desarrolladores, independientemente del trabajo que realice cada persona; no hay excepciones a esta regla.
- SCRUM no reconoce sub-equipos, no importan los dominios particulares que requieran ser tenidos en cuenta; no hay excepciones a esta regla.
- Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.

El tamaño óptimo del Equipo de Desarrollo es lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para completar una cantidad de trabajo significativa. Tener menos de tres miembros en el Equipo de Desarrollo reduce la interacción y resulta en ganancias de productividad más pequeñas, y tener más de nueve miembros en el equipo requiere demasiada coordinación. Los Equipos de Desarrollo de gran tamaño generan demasiada complejidad como para que pueda gestionarse mediante un proceso empírico. Los roles de *product owner* y Scrum Master no cuentan en el cálculo del tamaño del equipo a menos que también estén contribuyendo a trabajar en la Lista de Pendientes de la Iteración.

Roles auxiliares

Los clientes juegan un papel tan importante y activo dentro de la metodología SCRUM que son considerados como un rol participante. Son considerados parte del proceso también en algún nivel los usuarios finales del software e incluso los directivos de la empresa desarrolladora que no están involucrados directamente en el proceso.⁸

2.1.5 Eventos en SCRUM

La metodología SCRUM dicta que el Equipo SCRUM debe reunirse en diferentes ocasiones a lo largo del proyecto, los siguientes son los eventos más importantes que deben ocurrir en el proceso:

⁸ Schwaber, K. Advanced Development Methods. *SCRUM Development Process*. P 15

Iteración

El corazón de SCRUM son las iteraciones. Una iteración es un bloque de tiempo menor a un mes durante el cual se crea un incremento de producto utilizable, demostrable y potencialmente desplegable. Es más conveniente si la duración de las iteraciones es consistente a lo largo del proyecto. Cada nueva iteración comienza inmediatamente después de la finalización de la anterior.

Durante una iteración:

- No se realizan cambios que puedan afectar la meta de la iteración.
- Los objetivos de calidad nunca disminuyen.
- El alcance puede ser clarificado y renegociado entre el *product owner* y el Equipo de Desarrollo a medida que se va aprendiendo más.

Cada iteración puede considerarse un proyecto con un horizonte no mayor de un mes. Al igual que los proyectos, las iteraciones se usan para lograr algo específico. Cada iteración tiene una definición de qué se va a construir, un diseño y un plan flexible que guiará la construcción y el trabajo y el producto resultante.

Las iteraciones están limitadas a un mes calendario. Cuando el horizonte de una iteración es demasiado grande, la definición de lo que se está construyendo podría cambiar, la complejidad podría elevarse y el riesgo podría aumentar. Las iteraciones habilitan la predictibilidad al asegurar la inspección y adaptación del progreso al menos en cada mes calendario. Las iteraciones también limitan el riesgo al costo de un mes calendario.

Toda iteración contiene y consiste de las reuniones de Planeación de Iteración (Sprint Planning Meeting), Daily Scrums, el trabajo de desarrollo, la Revisión de la Iteración (Sprint Review), y la Retrospectiva de la Iteración (Sprint Retrospective), las cuales serán descritas a continuación:

Planeación de Iteración (Sprint Planning Meeting)

El trabajo a realizar durante la iteración se planifica en esta reunión. Este plan se crea mediante el trabajo colaborativo del Equipo SCRUM completo. La Reunión de Planeación de Iteración tiene un máximo de duración de ocho horas para una iteración de un mes, y para iteraciones más cortas, el evento es usualmente más corto. El Scrum Master es quien se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña al Equipo SCRUM a mantenerse dentro del bloque de tiempo.

La Reunión de Planeación de Iteración responde a las siguientes preguntas:

- ¿Qué puede entregarse en el Incremento resultante de la iteración que comienza?
- ¿Cómo se conseguirá hacer el trabajo necesario para entregar el Incremento?

Para la primera, el *product owner* discute el objetivo que la iteración debería lograr y los elementos del Listado de Producto que si se completan en la iteración lograrían la meta planteada. El Equipo SCRUM completo colabora en el entendimiento del trabajo. La entrada a esta reunión está constituida por el Listado del Producto, el último Incremento del producto y la capacidad y velocidad proyectada del Equipo de Desarrollo. El número de elementos de la Lista de Producto seleccionados para la iteración depende únicamente del Equipo de Desarrollo.

Solo el Equipo de Desarrollo puede evaluar qué es capaz de lograr durante la iteración que comienza. Después de que el Equipo de Desarrollo proyecta qué elementos del Listado del Producto entregará al final de la iteración, el Equipo SCRUM elabora una meta para la iteración. Esta meta debería lograrse durante la iteración a través de la implementación del Listado del Producto, y provee una guía al Equipo de Desarrollo de por qué se está construyendo el incremento.

Para la segunda, una vez establecida la meta y seleccionados los elementos del Listado, el Equipo de Desarrollo decide cómo construirá esta funcionalidad para formar un Incremento de producto. Este resultado en conjunto es lo que recibe el nombre de Lista de Pendientes de la Iteración.

El Equipo de Desarrollo por lo general comienza diseñando el sistema y el trabajo necesario para convertir la Lista de Producto en un Incremento de producto funcional: el trabajo podría ser de tamaño o esfuerzo estimado variables. Sin embargo, durante la Reunión de Planificación de la Iteración se planifica suficiente trabajo como para que el Equipo de Desarrollo pueda hacer una proyección de lo que cree que puede completar en la iteración que comienza.

Para el final de esta reunión, el trabajo planificado por el Equipo de Desarrollo para los primeros días de la Iteración es descompuesto en unidades de un día o menos llamadas tareas. El Equipo de desarrollo se auto-organiza para asumir el trabajo de la Lista de Pendientes de la Iteración, tanto durante la reunión de Planificación como a lo largo de la Iteración.

El *product owner* puede ayudar a clarificar los elementos del Listado del Producto seleccionados y hacer concesiones. Si el Equipo de Desarrollo determina que tiene demasiado trabajo o que no tiene suficiente trabajo, podría renegociar los elementos del Listado del Producto seleccionados con el *product owner*. El Equipo de Desarrollo podría también invitar a otras personas a que asistan con el fin de que proporcionen asesoría técnica o relacionada con el dominio. Al finalizar la

Reunión de Planeación de Iteración, el Equipo de Desarrollo debería ser capaz de explicar al *product owner* y al Scrum Master cómo pretende trabajar como un equipo autoorganizado para lograr la meta y crear el Incremento esperado.

Daily Scrum

El Daily Scrum es una reunión con un bloque de tiempo de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Daily Scrum y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente.

El Daily Scrum se realiza a la misma hora y en el mismo lugar todos los días para reducir la complejidad, si algún miembro del equipo no asiste a la reunión o no llega a tiempo, usualmente se le aplica algún tipo de castigo. Uno de los puntos más importantes para SCRUM es el de la transparencia: todos los miembros saben en todo momento que está haciendo el equipo, y los problemas deben ser sacados a la luz en cuanto se detectan para que el Scrum Master genere soluciones oportunas. Por esto, durante la reunión, cada miembro del Equipo de Desarrollo explica:

- ¿Qué hice ayer que ayudó al Equipo de Desarrollo a lograr la meta de la iteración?
- ¿Qué haré hoy para ayudar al Equipo de Desarrollo a lograr la meta de la iteración?
- ¿Veo algún impedimento que evite que el Equipo de Desarrollo o yo logremos la meta de la iteración?

El Equipo de Desarrollo usa el Daily Scrum para evaluar el progreso hacia la meta de la iteración y para evaluar qué tendencia sigue este progreso hacia la finalización del trabajo contenido en la Lista de Pendientes de la Iteración. Cada día, el Equipo de Desarrollo debería entender cómo intenta trabajar en conjunto como un equipo auto-organizado para lograr la meta de la iteración y crear el Incremento esperado hacia el final de ésta. El Equipo de Desarrollo o los miembros del equipo a menudo se vuelven a reunir inmediatamente después del Daily Scrum para tener discusiones detalladas o para adaptar o replanificar el resto del trabajo de la iteración.

El Scrum Master se asegura de que el Equipo de Desarrollo tenga la reunión, pero el Equipo de Desarrollo es el responsable de dirigir el Daily Scrum. El Scrum Master enseña al Equipo de Desarrollo para que mantenga el Daily Scrum en los límites del bloque de tiempo de 15 minutos.

El Scrum Master se asegura de que solo los miembros del Equipo de Desarrollo participan en la reunión. Los Daily Scrums mejoran la comunicación, eliminan la necesidad de mantener otras reuniones, identifican y eliminan impedimentos relativos al desarrollo, resaltan y promueven la toma de decisiones rápida, y mejoran el nivel de conocimiento del Equipo de Desarrollo. El Daily Scrum

constituye una reunión clave de inspección y adaptación.

Revisión de Iteración (Sprint Review)

Al final de la iteración se lleva a cabo una Revisión de Iteración para inspeccionar el Incremento y adaptar el Listado del Producto si fuese necesario. Durante la Revisión de Iteración, el Equipo SCRUM y los interesados colaboran acerca de lo que se hizo durante la iteración. Basándose en esto, y en cualquier cambio al Listado del Producto a lo largo de la iteración, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor. Se trata de una reunión informal, no una reunión de seguimiento, y la presentación del Incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración.

Se trata de una reunión restringida a un bloque de tiempo de cuatro horas para iteraciones de un mes; para iteraciones más cortas, se reserva un tiempo proporcionalmente menor. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña a todos a mantener el evento dentro del bloque de tiempo fijado. Cualquier trabajo que no esté completo o no tenga entregable no es tomado en cuenta.

El resultado de la Revisión de Iteración es un Listado del Producto revisado, que define los posibles elementos para la siguiente iteración. Es posible además que el Listado del Producto reciba un ajuste general para enfocarse en nuevas oportunidades.

Retrospectiva de la iteración (Sprint Retrospective)

El objetivo de esta reunión es identificar las fortalezas y debilidades de la iteración realizada por el equipo; es una oportunidad para que el Equipo SCRUM se inspeccione a sí mismo y cree un plan de mejoras para ser abordadas durante la siguiente iteración. La Retrospectiva de la Iteración tiene lugar después de la Revisión de iteración y antes de la siguiente Reunión de Planificación. Se trata de una reunión restringida a un bloque de tiempo de tres horas para iteraciones de un mes. Para iteraciones más cortas se reserva un tiempo proporcionalmente menor. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito, enseña a todos a mantener el evento dentro del bloque de tiempo fijado y participa en la reunión como un miembro del equipo ya que la responsabilidad del proceso SCRUM recae sobre él.

El propósito de la Retrospectiva de Iteración es:

- Inspeccionar cómo fue la última iteración en cuanto a personas, relaciones, procesos y herramientas.
- Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras.
- Crear un plan para implementar las mejoras a la forma en la que el Equipo SCRUM desempeña su trabajo.

El Scrum Master alienta al equipo para que mejore dentro del marco de proceso SCRUM, su proceso de desarrollo y sus prácticas para hacerlos más efectivos en la siguiente iteración. Durante cada Retrospectiva, el Equipo SCRUM planifica formas de aumentar la calidad del producto mediante la adaptación de la Definición de “Terminado” según sea conveniente.

Para el final de la Retrospectiva de iteración, el Equipo SCRUM debería haber identificado mejoras que implementará en la próxima iteración. El hecho de implementar estas mejoras constituye la adaptación subsecuente a la inspección del Equipo de Desarrollo a sí mismo. Aunque las mejoras pueden implementarse en cualquier momento, la Retrospectiva de Iteración ofrece un evento dedicado para este fin, enfocado en la inspección y la adaptación.

2.1.6 Documentos en SCRUM

Todo evento realizado dentro de la metodología SCRUM trae consigo una entrada y una salida documentable. Las siguientes secciones describen cada uno de los documentos escritos que reflejan el avance de un proyecto SCRUM.

User Stories (Historias de usuario)

Aunque no hacen parte directamente de la metodología SCRUM, representan el punto de partida del proyecto. Son el resultado de la reunión del *product owner* con el cliente y son las funcionalidades que éste espera ver en su software descritas desde un punto de vista alejado de la perspectiva del equipo de desarrollo. La guía para desarrolladores de Microsoft define una historia de usuario de la siguiente manera:

“Una historia de usuario comunica la funcionalidad que implica algún valor para el usuario final del producto o sistema a desarrollar. Cada historia de usuario debe describir simplemente lo que un usuario quiere que el software haga descrito desde su punto de vista. Cuando se escriben historias de usuario, debería enfocarse en para quien es la funcionalidad descrita, que esperan de ella, y

porqué. Se debe evadir cualquier descripción que especifique como debería desarrollarse la funcionalidad descrita⁹.

Las historias de usuario deben cumplir con el principio INVEST de Bill Wake¹⁰:

- *Independent (Independiente)*: la historia de usuario no debe depender de otra historia ya que esto facilitará la priorización de las mismas. Si por alguna razón la historia de usuario es dependiente se recomienda fusionarlas.
- *Negotiable (Negociable)*: la historia de usuario puede cambiar y evolucionar a lo largo de la ejecución del proyecto, incluso podría dejar de tenerse en cuenta si así el cliente lo desea.
- *Valuable (Valiosa)*: la historia de usuario debe brindar valor al proyecto y al usuario final.
- *Estimable (Estimable)*: la historia de usuario debe tener el tiempo que ésta tomará en implementarse.
- *Small (Pequeña)*: la historia de usuario debe ser pequeña y concisa. Si una historia de usuario es muy grande ésta se debe dividir en otras historias más pequeñas, esto con el fin de poder tener un mejor control sobre ellas.
- *Testable (Comprobable)*: la historia de usuario debe poderse probar en un proceso de calidad.

Un ejemplo de historia de usuario podría ser el siguiente:

“Cuando se desee hacer un pago electrónico por nuestro producto, deberíamos aceptar solo Visa, MasterCard y American Express. Para compras superiores a 200.000 pesos, se debe consultar por el código de seguridad en la parte posterior de la tarjeta. El sistema debe poder averiguar el tipo de tarjeta a partir de los dos primeros números del número de identificación. El número de tarjeta usado debería quedar consignado para usos futuros del mismo usuario, además de la fecha de expiración de la tarjeta”.

Product Backlog (Lista o Listado del Producto)

El Listado del Producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los

⁹ Microsoft Developer Network. *Artifacts, Work items and Workflow, User Story, Defining a User Story*.

¹⁰ INVEST in Good Stories, and SMART Tasks (17-08-2003). Bill Wake. XP123.

elementos de la Lista de Producto tienen como atributos la descripción, la ordenación, la estimación y el valor. Es responsabilidad del *product owner* generar este listado según las historias de usuario resultantes de su interacción con el cliente y asignar prioridades y descripciones. Corresponde al Equipo de Desarrollo escoger los estimados y entregables para cada elemento del Listado del Producto.

Un Listado de Producto nunca está completo, el desarrollo más temprano de éste solo refleja los requisitos conocidos y mejor entendidos al principio del proyecto. El listado evoluciona a medida de que el producto y el entorno en el que se usará también lo hacen. El Listado de Producto es dinámico; cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil. Mientras el producto exista, su Listado de Producto también existe y seguirá cambiando. A menudo, varios Equipos SCRUM trabajan juntos en el mismo producto. Cuando esto ocurre, para describir el trabajo a realizar sobre el producto, se utiliza un único Listado de Producto.

Añadir o editar detalles, estimaciones y prioridades a los elementos de la Lista del Producto se trata de un proceso continuo, en el cual el *product owner* y el Equipo de Desarrollo colaboran acerca de los detalles variables de los elementos del Listado del Producto. Durante este refinamiento, se examinan y revisan sus elementos, el equipo SCRUM decide cómo y cuándo se hace el refinamiento. Este usualmente consume no más del 10% de la capacidad del Equipo de Desarrollo. Sin embargo, los elementos de la Lista de Producto pueden actualizarse en cualquier momento por el *product owner* o a criterio suyo. El Equipo de Desarrollo es el responsable de proporcionar todas las estimaciones. El *product owner* podría influenciar al Equipo ayudándoles a entender y seleccionar soluciones de compromiso, pero las personas que harán el trabajo son las que deben hacer la estimación final.

En métodos tradicionales de manejo de proyectos, los administradores tienden a determinar la capacidad de trabajo de un miembro del Equipo de Desarrollo en términos de tiempo; es decir, estiman el tiempo que podrían tomar las que tareas que se llevarán a continuación y asignan trabajo basado en el total de tiempo disponible de ese miembro del equipo. Esto es problemático, ya que no hay una forma confiable de distinguir entre un trabajo que es muy difícil de completar y otro que es poco exigente teniendo sólo en cuenta una estimación del tiempo que va a tomar el trabajo. Para decirlo de otro modo, codificar una funcionalidad y organizar los documentos en el escritorio son actividades que probablemente tomen la misma cantidad de tiempo, pero no hay duda de que el primero requiere una mayor concentración y esfuerzo. Debido a ese hecho, deberían tratarse como diferentes tareas, pues requieren diferentes niveles de esfuerzo.

SCRUM adopta un enfoque muy diferente a la determinación de la capacidad de un miembro del equipo. En primer lugar, SCRUM asigna trabajo a todo un equipo, no a un miembro individual, poniendo énfasis en el esfuerzo colectivo. En segundo lugar, los Equipos SCRUM prefieren no cuantificar el trabajo en términos de tiempo, ya que esto socavaría la auto-organización que es uno de los pilares de la metodología. Se trata de una ruptura importante frente a los métodos tradicionales de manejo de proyectos.

¿Cómo funciona entonces el método de estimación? SCRUM no establece un único camino para que los equipos puedan estimar su trabajo; sin embargo, sí se pide que los equipos nunca utilicen términos de tiempo y busquen una métrica más relativa para cuantificar el esfuerzo. Métodos de estimación numérica comunes incluyen tamaño (1 a 10), talla de ropa (XS , S , M , L , XL , XXL , XXXL), la secuencia de Fibonacci (1 , 2 , 3 , 5 , 8 , 13 , 21 , 34...), e incluso las razas de perros, en el que un Chihuahua podría representar las historias más pequeñas y un gran danés las más grandes. Lo importante es que el equipo comparta una comprensión de la escala que se utiliza y que cada miembro del equipo se sienta cómodo con los valores de la escala.

En la Reunión de Planificación de Iteración, el equipo se sienta a valorar su esfuerzo para las historias en el listado del producto. El *product owner* necesita estas estimaciones, pues él o ella tiene la facultad de priorizar eficazmente los elementos del Listado del Producto y como consecuencia, ordenar las próximas iteraciones según la capacidad del Equipo de Desarrollo. Esto significa que el propietario del producto necesita una evaluación honesta de que es realmente un trabajo difícil. Incluso cuando el equipo estima entre sí, se deben tomar medidas para evitar influenciar las estimaciones del equipo. Como tal, se recomienda que todos los miembros del equipo revelen sus estimaciones de forma simultánea. Sin embargo, incluso cuando los equipos poseen una comprensión compartida de su escala, no pueden dejar de estimar de manera diferente y se requieren múltiples rondas para llegar a un consenso.

***Sprint Backlog* (Lista de Pendientes de la Iteración)**

Es el resultado de la Reunión de Planeación de Iteración. Corresponde al conjunto de elementos de la Lista del Producto seleccionados para la iteración, más un plan para entregar el Incremento de producto y conseguir la meta planteada para la iteración. La Lista de Pendientes de la Iteración hace visible todo el trabajo que el Equipo de Desarrollo identifica como necesario para alcanzar la meta. 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 "Terminado".

Aunque esto varía significativamente para cada Equipo SCRUM, los miembros del Equipo deben tener un entendimiento compartido de lo que significa que el trabajo esté completado, para asegurar la transparencia. Esta es la definición de “Terminado” para el Equipo SCRUM y se utiliza para evaluar cuándo se ha completado el trabajo sobre el Incremento de producto.

Esta misma definición guía al Equipo de Desarrollo en saber cuántos elementos de la Lista de Producto puede seleccionar durante una reunión de Planificación de Iteración. El propósito de cada iteración es entregar incrementos de funcionalidad que potencialmente se puedan poner en producción, y que se ajustan a la Definición de “Terminado” actual del Equipo SCRUM.

La Lista de Pendientes de la Iteración es un plan con un nivel de detalle suficiente como para que los cambios en el progreso se puedan entender en el Daily Scrum. El Equipo de Desarrollo modifica la Lista de Pendientes de la Iteración a lo largo de la iteración y éste evoluciona a lo largo de la iteración. Esto ocurre a medida que el Equipo de Desarrollo trabaja sobre el plan y aprende más acerca del trabajo necesario para conseguir la meta planteada.

Según se requiere nuevo trabajo, el Equipo de Desarrollo lo añade a la Lista de Pendientes de la Iteración en forma de tareas. A medida que el trabajo se ejecuta o se completa, se va actualizando la estimación de trabajo restante. Cuando algún elemento del plan pasa a ser considerado innecesario, es eliminado. Solo el Equipo de Desarrollo puede cambiar su Lista de Pendientes de la Iteración durante la iteración. La Lista de Pendientes de la Iteración es una imagen visible en tiempo real del trabajo que el Equipo de Desarrollo planea llevar a cabo durante la iteración, y pertenece únicamente al Equipo de Desarrollo.

Burndown Chart (Gráfica de avances)

En cualquier momento, es posible sumar el trabajo total restante para alcanzar el objetivo del proyecto. El *product owner* hace seguimiento de este trabajo restante total al menos en cada Revisión de Iteración, comparando esta cantidad con el trabajo restante en revisiones previas, para evaluar el progreso hacia la finalización del trabajo proyectado en el tiempo deseado para el objetivo. Esta información se muestra de forma transparente a todos los interesados.

La Burndown Chart o gráfica de avance es un método que permite monitorear el rendimiento del equipo según las metas planteadas. Según su comportamiento es fácil identificar si el equipo tiene en determinado momento pocos o muchos elementos del listado asignados para su ritmo de trabajo. La gráfica dibuja en un plano bidimensional (Días vs puntos de historia finalizados) la recta ideal que el proyecto debería cumplir para terminar todos los puntos de historia planteados en el

tiempo estimado inicialmente; sobre esta ubica los datos reales que se actualizan cada vez que un ítem del listado del producto es finalizado.¹¹

Estas gráficas, aunque útiles e informativas, no reemplazan la importancia del empirismo. En entornos complejos, siempre se desconoce lo que ocurrirá mas adelante en el proyecto, así que solo lo que ya ha ocurrido puede utilizarse para la toma de decisiones con miras al futuro.

¹¹ *Palacio, Juan. Flexibilidad con Scrum: Principios de diseño e implantación de campos de Scrum. p 140*

2.2 Tecnologías aplicadas

La aplicación se desarrollará en arquitectura web (cliente servidor), donde de lado del servidor se utilizará PHP como lenguaje y del lado del cliente las conocidas herramientas web para desarrollar correctamente los objetivos.

Las herramientas a utilizar son las siguientes:

2.2.1 Servidor:

MySQL

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

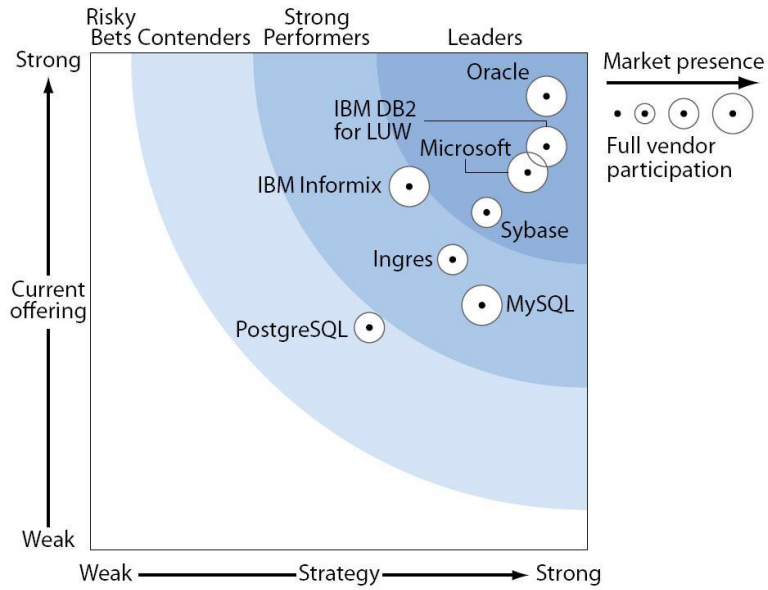
El diseño completo de la base estará realizado bajo un software especializado realizado por la misma comunidad que apoya MySQL¹².

Por qué Mysql

En primer lugar, economía. Se está realizando un proyecto con productos OpenSource o productos licenciados libremente por los fabricantes. Seguramente si el proyecto fuera de mayor tamaño, se elegiría una opción de base de datos propietaria, como Oracle (que es para muchos la mejor opción por estudios, prácticas o consenso) según el estudio comparativo de bases de datos realizado por Forrester¹³.

¹² Página oficial de MySQL, <http://www.mysql.com/>

¹³ The Forrester Wave™: Enterprise Database Management Systems, Q2 2009. Pág. 7



En segundo lugar, se requieren productos con reconocido prestigio, fiabilidad, velocidad, rendimiento, facilidad de administración y conexión con otros productos, bien documentados, con una buena evolución y soporte. Productos de los que sea fácil obtener información, con buenas herramientas, y para los que incluso se puedan recibir cursos de formación si fuese necesario. Productos que estén siendo utilizados en muchos entornos productivos y que nos den la suficiente confianza.

Además, los gestores de bases de datos OpenSource hace tiempo que dejaron de ser un experimento y ya son una alternativa real para las empresas. Son productos cada vez más evolucionados, con más funcionalidades y las empresas que los desarrollan tienen también cada vez más volumen de negocio (cuestión importante para continuar la evolución de los productos).

MySQL Workbench

Es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL. Es el sucesor de DBDesigner 4 de fabFORCE.net, y reemplaza el anterior conjunto de software, MySQL GUI Tools Bundle.

Por qué MySQL Workbench

Para la funcionalidad de la página del lado del servidor, se buscaron herramientas fáciles de implementar y con mucho apoyo por parte de la comunidad con el fin de agilizar el uso de herramientas, con lo que fue seleccionado PHP.

PHP

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.¹⁴

Sobre php se utilizará un framework (marco de trabajo) para facilitar y agilizar el trabajo con el fin de mejorar las técnicas de desarrollo.

Por qué PHP

Por cuatro grandes características: Velocidad, estabilidad, seguridad y simplicidad.

Velocidad: No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.

Estabilidad: La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para lo bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

Seguridad: El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini

¹⁴ Página oficial de PHP, <http://www.php.net/index.php>

Simplicidad: Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

Bueno otra característica a tener en cuenta sería la conectividad. PHP dispone de una amplia gama de librerías, y agregarle extensiones es muy fácil. Esto le permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras.

Codeigniter

Codeigniter es un framework PHP potente con un tamaño muy pequeño, construido para programadores PHP que necesitan una herramienta simple y elegante para crear aplicaciones web con todas las funciones¹⁵.

Por qué Codeigniter

La preferencia de este framework es la cantidad de apoyo en las redes y la popularidad que maneja; además cuenta con documentación actualizada y referenciada con ejemplos para fácil comprensión de todo lo que allí se menciona¹⁶.

En general un framework se utiliza para tener ventajas en:

- La seguridad es un factor clave: datos introducidos por el usuario debidamente validados y filtrados para evitar ataques XSS.
- Comunicación con Base de Datos automatizada, validando todas las consultas y filtrando los datos variables para evitar inyección SQL.
- Disponer de componentes plenamente probados, con el objetivo de mejorar la productividad.

2.2.2 Cliente:

HTML

Siglas de HyperText Markup Language («lenguaje de marcado hipertextual»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el

¹⁵ *Página oficial de Codeigniter*, <http://ellislab.com/codeigniter>

¹⁶ *Manual de usuario de Codeigniter*, <http://ellislab.com/codeigniter/user-guide/>

texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir o hacer referencia a un tipo de programa llamado script, el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.¹⁷

CSS

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.¹⁸

Teniendo en cuenta que la funcionalidad no solo va del lado del cliente, existen varias formas de conectar el servidor con el cliente de forma más amigable para el navegador web. Entre agregados de terceros o herramientas que trae por defecto el navegador, se decidió utilizar lo más compatible con la mayoría de los navegadores.

Javascript

Javascript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

No es conveniente utilizar Javascript puro debido a la alta probabilidad de fallos que puede tener programar en este lenguaje. Por ello, por la gran cantidad de documentación y manejo de excepciones que tiene, se seleccionó un conjunto de 37 items de trabajo en javascript.

Jquery

Es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos,

17 Definición tomada de la red <http://www.ri5.com.ar/ayuda03.php>

18 Definición tomada de la página oficial de w3c, <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>

desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.¹⁹

Para evitar errores en el momento de visualizar las páginas (Cada navegador web es libre de usar un interpretador de HTML, CSS y Javascript independiente) se utilizará un framework de trabajo rápido en diseño de vistas.

Por qué JQuery

- Funciones con efectos visuales e interacciones dinámicas (Ajax, auto-completar, elementos que aparecen/desaparecen, animaciones y mucho más.).
- Compatibilidad total en los navegadores más comunes: Internet explorer 6 y superiores, Firefox, Chrome, Safari, Opera.
- Si lo se utiliza correctamente no trae ningún problema con respecto a la indexación de los motores de búsqueda.

Twitter Bootstrap

Es una colección de herramientas de software libre para la creación de sitios y aplicaciones web. Contiene plantillas de diseño basadas en HTML y CSS con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz, así como extensiones opcionales de JavaScript.

Este framework es de los más populares que se pueden encontrar en la red ya que contiene muchos agregados de desarrollo libre aceptados por muchos desarrolladores expertos en el tema. Actualmente muchas páginas han sido desarrolladas en este framework. Actualmente páginas que estén trabajadas sobre el framework incluyen la NASA y la MSNBC.

Por qué Twitter Bootstrap

Ventajas

- Bootstrap es fruto de la propia comunidad web quién ha creado y aportado nuevas metodologías de trabajo al desarrollo front-end. Además utiliza servicios y componentes creados por la comunidad como HTML5 shim, Normalize.css, OOCSS, jQuery UI (implementa 12 plugins), LESS y GitHub.

19 *Diagrama estadístico de uso de bibliotecas basadas en javascript,*
http://w3techs.com/technologies/overview/javascript_library/all

- Los conceptos que ha insertado Bootstrap al desarrollo front-end han resultado un compendio de buenas prácticas que perdurarán en el tiempo.
- Tiene una enorme comunidad que soporta este desarrollo y cuenta con implementaciones externas como WordPress, Drupal, SASS o jQuery UI.
- Como se ha dicho anteriormente una de sus ventajas es la facilidad de construir sitios rápidamente, y prototipar sitios webs y aplicaciones con un bajo coste.
- La última ventaja es que Bootstrap trae un theme por defecto bastante optimizado. Obviamente modificable, o bien, puedes comprar temas ya hecho en Wrapp Bootstrap.

Desventajas

- Aunque la curva de aprendizaje es liviana, sí que es verdad que para poder empezar a trabajar con Bootstrap debes familiarizarte con la estructura y con la nomenclatura.
- Debes adaptar tu diseño gráfico a un grid de doce columnas (por defecto), y el ancho de las columnas y el gutter se modifica según el dispositivo para el que vayas a trabajar y su contexto.
- Es complicado (o casi imposible) cambiar de versión si has realizado modificaciones profundas sobre el core.
- Si necesitas añadir componentes que no existen, quizá debas personalizarlos en CSS para que mantenga la coherencia con tu diseño y cuidar el responsive.

2.3 ESTADO DEL ARTE

Existen diferentes soluciones para realizar SCRUM basadas en herramientas web, la tendencia actual consiste en mantener la información actualizada en la nube con el fin de acceder a ella en cualquier momento y a través de usuarios que representen los diferentes roles del equipo SCRUM. Se han seleccionado como referencias del estado del arte herramientas software que representan tres niveles diferentes de tamaño y costo, siendo OnTime la más completa y costosa y Scrumy la más asequible.

2.3.1 OnTime Scrum:

<http://www.ontimenow.com/>

Basado en la tecnología HTML5. Administra la cartera de su Equipo de Desarrollo, iteraciones y desarrollos y puede manejar un número ilimitado de proyectos. Maneja un tablero de elementos, un tablero de planeación y cuenta con estadísticas basadas en el desarrollo. En el sistema se adiciona un sistema de cartas que permiten observar de mejor manera las historias de usuario para poder moverlas con facilidad. Mantiene una integración con los IDE de desarrollo usados, con el fin de siempre mantener actualizado el sistema.

2.3.2 Scrumwise:

<http://www.scrumwise.com/>

Basado en tecnología HTML5, también desarrolla iteraciones y mantiene al tanto los grupos de trabajo con un número limitado de proyectos según el pago de la herramienta por mensualidad. Esta se basa mucho más en el Daily Scrum y sus estadísticos para mantener informado al Scrum Master. El tablero de control mantiene mucho más cercana la idea de “stickers” que en la metodología se menciona con bastante énfasis.

2.3.3 Scrumy

www.scrumy.com

Scrumy es la solución sencilla, económica y accesible basada en la web para todo aquel que desee una guía básica y simple para implementar SCRUM. A pesar de tener una funcionalidad estrictamente enfocada en el tablero de Daily Scrum y la gráfica de rendimiento, se ha establecido como una herramienta reconocida en el ámbito del desarrollo ágil de software.

2.3.4 Cuadro comparativo de funcionalidades:

	OnTime	ScrumWise	Scrumy	Proyecto
Listado del Producto	X	X	X	X
Dividir PBI				X
Cantidad de PBIS a lo largo del proyecto				X
Equipo SCRUM	X	X		X
División por roles SCRUM		X		
Velocidad del equipo	X	X		X
Cambios del equipo por iteración				X
Información de PBIs	X	X		X
Estimación de tiempo	X	X	X	
Estimación por puntos de historia		X	X	X
Encargado	X	X	X	
Creación de atributos extra	X			
Prioridad numérica				X
Iteraciones y Entregables	X	X		X
Verificar prioridades de PBIs				X
Meta				X
Tablero de Daily Scrum	X	X	X	X
Creación de estados extra	X			
Porcentaje de PBI finalizado	X	X		X
Gráficas de Rendimiento	X	X		X
Notificaciones y alertas	X	X		
Varios equipos por proyecto	X	X		
Manejo de días no hábiles	X	X		
Multiusuario	X	X		
Seguimiento de Trabajo en tiempo real	X	X		

3. FASE PRELIMINAR DE SCRUM

La fase preliminar del proyecto incluye todos los eventos realizados antes de comenzar la primera iteración. Esto incluye reunión con el cliente, creación del Listado del Producto, estimación de niveles de esfuerzo para elementos en el Listado, estimación de duración de proyecto, establecimiento de duración de iteraciones y Planeación de la primera Iteración del proyecto.

3.1 Historias de Usuario (User Stories)

Al reunirse con las partes interesadas, el *product owner* registra lo que vendría a ser la funcionalidad que se espera del sistema a desarrollar desde un punto de vista netamente de usuario final y alejado del entorno de desarrollo. Para nuestro proyecto, el usuario final es el Scrum Master, rol explicado ampliamente en la sección de marco teórico y quien tiene por responsabilidad primaria, verificar que la metodología SCRUM sea correctamente implementada en su equipo. Estas serían las historias de usuario para nuestro proyecto:

- “Como Scrum Master quisiera una herramienta que me diera soporte de documentación y retroalimentación constante del proyecto que estoy asesorando para así tener más tiempo disponible para dedicarle a mi equipo. También debería poder identificar la fase de la metodología SCRUM en que se encuentra el proyecto y cubrir cada reunión y documentación de las mismas. Esto incluye poder crear y administrar para cada proyecto: listado del producto, Equipo de Desarrollo, iteraciones, Daily Scrums, tareas y gráfica de rendimiento”.
- “Para el Listado del Producto, Quiero poder guardar cada elemento con sus respectivos atributos y cambiarlos como me parezca mientras no estén vinculados a una iteración. Cada uno de estos debe ser totalmente editable, eliminable y sería excelente si tuviera la opción de dividirse también. El listado debe ordenarse por prioridad, y solo los elementos mas prioritarios deberían poderse agregar a una iteración. Sería excelente si el sistema me mostrara las cantidades de elementos del Listado del Producto en el sistema según su estado para llevar un control de nuestras capacidades de planeación y nuestros fallos al crear y priorizar elementos del listado”.

- “El sistema debería calcular la cantidad de iteraciones y daily scrums que tendrá el proyecto en base a los estimados de tiempo que se generan en la fase preliminar. Sin embargo, debería permitir crear mas iteraciones en el caso muy probable de que los estimados no sean precisos. Todas las iteraciones deben tener definida una meta, el lugar y la hora de los Daily Scrums y una fecha de entrega del incremento”.
- “Quiero que el sistema actualice por si mismo las gráficas de rendimiento y velocidad del Equipo de Desarrollo en base a los elementos del Listado del Producto que vayan siendo terminados y sus respectivos niveles de esfuerzo. Sería muy bueno que el sistema diera consejos sobre la cantidad de ítems a agregar en una iteración en base a la capacidad promedio del equipo hasta el momento e hiciera advertencias si la cantidad de elementos del Listado del Producto finalizados no corresponde con los estimados iniciales”.
- “Quiero poder guardar el estado del tablero de proyecto para cada Daily Scrum y poder revisar cualquiera de estos en cualquier momento del proyecto. Quiero que sea tan fácil de actualizar como el mismo tablero de sticky notes en la oficina y que el proceso de crear tareas nuevas y moverlas a través del tablero no me gaste mas tiempo que el proceso físico”.

3.2 Product Backlog (Listado del Producto)

Después de registradas las historias de usuario, es trabajo del *product owner* descifrar el lenguaje de usuario final y convertirlas en elementos del Listado del Producto, los cuales serán llevados a cabo por el Equipo de Desarrollo para generar incrementos del software. Un elemento o ítem del Listado del Producto debe estar debidamente descrito para que todos los miembros del Equipo de Desarrollo entiendan lo que se debe hacer.

El *product owner* da prioridad a cada 44tem dependiendo de que tan importante ha identificado que este podría ser para el cliente, y el Equipo de Desarrollo le da una valoración al esfuerzo que requerirá terminarlo. A lo largo del proyecto, los elementos del listado suelen variar muchas veces, pueden ser eliminados, editados o hasta divididos.

Listado del Producto Fase Preliminar:

ID	1
NOMBRE	Fase Inicial de Proyecto
DESCRIPCIÓN	El usuario crea una nueva cuenta, nuevo proyecto, grupo de trabajo y listado completo de ítems del producto (priorizados y descritos). La información consignada durante esta fase debe ser totalmente editable hasta comenzada la primera iteración. No pueden crearse ítems con la misma prioridad que uno existente.
PRIORIDAD	200

ID	2
NOMBRE	Fase Preliminar SCRUMs
DESCRIPCIÓN	El usuario asigna puntos de historia y entregable a cada ítem del listado del producto y selecciona la duración que tendrán las iteraciones y el proyecto en total. La información consignada durante esta fase debe ser totalmente editable hasta comenzada la primera iteración.
PRIORIDAD	180

ID	4
NOMBRE	DAILY SCRUM
DESCRIPCIÓN	Visualización y edición de estados de los ítems a desarrollar en la iteración y sus respectivas tareas (no empezada, en progreso, finalizada), agregar tareas no planeadas al listado, dividir ítems del listado en ítems más pequeños. No pueden crearse ítems con la misma prioridad que uno existente. La información consignada en el daily scrum debe ser totalmente editable hasta ser confirmada mediante un botón. Una tarea entregable no puede ser finalizada mientras existan otras tareas de ese 44tem en progreso. Si un ítem no ha sido finalizado en el último daily scrum de la iteración, debe regresar al listado de producto.
PRIORIDAD	160

ID	3
NOMBRE	Planeación de iteración

DESCRIPCIÓN	Definir meta, cambios en el equipo de trabajo, ítems a desarrollar, fecha de entregable y lugar y hora de DAILY SCRUM para la iteración. Solo pueden asignarse a una iteración los ítems de mayor prioridad en el listado del producto. Si un nuevo miembro del equipo es creado en este momento, es vinculado automáticamente a la iteración actual.
PRIORIDAD	120

ID	5
NOMBRE	Retroalimentación de proyecto
DESCRIPCIÓN	Visualización de información del proyecto, que incluye los datos consignados en cada iteración y daily scrum, las gráficas de rendimiento y velocidad del equipo y la información de ítems creados a lo largo del proyecto.
PRIORIDAD	100

3.3 Estimación de Entregable y Niveles de Esfuerzo

Finalizada la labor del *product owner* al crear el Listado del Producto, corresponde al Equipo de Desarrollo analizar cada uno de los elementos allí consignados para establecer un entregable que represente su definición de “Terminado” y los niveles de esfuerzo necesario para que el Equipo logre terminar completamente cada uno de estos. Se decidió como entregable para cada elemento el desarrollo de un módulo web que cumpliera con la funcionalidad explicada en el atributo de descripción de cada elemento. Para los niveles de esfuerzo, Inicialmente se utilizó una medición de tiempo en días, dando como resultado los siguientes estimados:

ID	1
NOMBRE	Fase Inicial de Proyecto
TIEMPO ESTIMADO	10 días

ID	2
NOMBRE	Fase Preliminar SCRUMs
TIEMPO ESTIMADO	5 días

ID	3
NOMBRE	Planeación de iteración
TIEMPO ESTIMADO	5 días

ID	4
NOMBRE	DAILY SCRUM
TIEMPO ESTIMADO	30 días

ID	5
NOMBRE	Retroalimentación de proyecto
TIEMPO ESTIMADO	5 días

Sin embargo, a través de la investigación de mejores prácticas en desarrollo ágil, se decidió relativizar los estimados temporales a otro sistema para minimizar riesgos y realizar una mejor planeación. Por este motivo se decidió aplicar el método de valoración por puntos de historia, asumiendo valores relativos de esfuerzo que oscilan desde 1 (menor esfuerzo: 3-5 días), hasta 6 (20-30 días).

ID	1
NOMBRE	Fase Inicial de Proyecto
PUNTOS DE HISTORIA	2

ID	2
NOMBRE	Fase Preliminar SCRUMs
PUNTOS DE HISTORIA	1

ID	3
NOMBRE	Planeación de iteración
PUNTOS DE HISTORIA	1

ID	4
NOMBRE	DAILY SCRUM
PUNTOS DE HISTORIA	6

ID	5
NOMBRE	Retroalimentación de proyecto
PUNTOS DE HISTORIA	2

3.4 Duración de Proyecto e Iteraciones

Haciendo uso de los estimados del Equipo de Desarrollo es posible plantear las fechas y los intervalos que habrá entre cada incremento demostrable del proyecto. Estableciendo una capacidad de desarrollo del Equipo de tres puntos de historia por iteración, se decidió manejar iteraciones de diez días hábiles con dos días de intermedio entre ellas para realizar los eventos de Revisión, Retrospectiva y Planeación de Iteraciones. Siendo el total de puntos de historia del proyecto doce, se establecieron cuatro iteraciones para el desarrollo de la herramienta; generando un estimado inicial de dos meses (50 días hábiles) de desarrollo hasta el incremento final.

3.5 Planeación de Primera Iteración

Para la primera iteración se escogieron los elementos del Listado del Producto mas prioritarios hasta alcanzar la capacidad del Equipo de Desarrollo. Como resultado, se añadieron a la iteración los elementos “Fase inicial del Proyecto” y “Fase Preliminar SCRUMs”, para los que se generó un listado de tareas asociadas como guía para el desarrollo del entregable. La hora del Daily Scrum se estableció para las 10 am en el lugar de trabajo y se fijó como meta de la iteración “Desarrollar un incremento que permita al usuario documentar la Fase Preliminar de los proyectos SCRUM vinculados a su cuenta”.

Lista de Pendientes de la Primera Iteración:

Iteración 1		
PBI	1	Fase Inicial de Proyecto
-		Diseñar y programar vistas.
-		Validar inicio de sesión.
-		Crear nuevo usuario.
-		Verificar al crear usuario que no tenga un correo existente.
-		Crear nuevo proyecto.
-		Cargar información de proyecto en variable de sesión.

- Validar existencia de miembros del equipo.
- Validar existencia de ítems en el listado.
- Crear nuevo miembro del equipo.
- Crear nuevo ítem en el listado.
- Verificar al crear un nuevo ítem que no tenga una prioridad existente.
- Editar miembro del equipo.
- Editar ítem del listado.
- Validar campos vacíos y numéricos en formularios.

PBI	2	Fase Preliminar SCRUMs
-----	---	------------------------

- Diseñar y programar vistas.
- Validar precisión de ítems.
- Validar duración de proyectos e iteración.
- Precisar ítem (agregar puntos de historia y entregable).
- Generar tarea entregable.
- Establecer duración de proyecto e iteración.
- Editar duración de proyecto e iteración.
- Editar precisión de ítem.
- Validar campos vacíos y numéricos en formularios.
- Aplicar multiplicadores (día, semana, mes) a duración de proyecto.

4. DESARROLLO E IMPLEMENTACIÓN

SCRUM basa su metodología en un desarrollo incremental e iterativo al entender la naturaleza variable e inestable que puede llegar a presentar un proyecto de software. Si bien los eventos descritos en la Fase Preliminar permiten generar estimados de tiempo y esfuerzo para el desarrollo del Proyecto, es muy improbable que el mismo se realice sin variantes a lo largo de cada una de las iteraciones planteadas. SCRUM facilita y promueve el proceso de adaptación al cambio, mientras brinda herramientas que permiten al equipo permanecer dentro de la meta minimizando los costos de tiempo y esfuerzo. En esta sección se describe el trabajo realizado a lo largo del proyecto, y los cambios ocurridos respecto a la planeación inicial.

4.1 Cambios realizados en el Listado del Producto

Siendo una mala práctica dentro de la metodología SCRUM manejar niveles de esfuerzo para un elemento del Listado del Producto superiores a los permitidos por una iteración, se dividió el elemento “Daily Scrum” en dos partes de tres puntos de historia para ajustarse de esa forma a la planeación inicial.

ID	4
NOMBRE	DAILY SCRUM
DESCRIPCIÓN	Visualización y edición de estados de los ítems a desarrollar en la iteración y sus respectivas tareas (no empezada, en progreso, finalizada), agregar tareas no planeadas al listado, dividir ítems del listado en ítems más pequeños. No pueden crearse ítems con la misma prioridad que uno existente. La información consignada en el daily scrum debe ser totalmente editable hasta ser confirmada mediante un botón. Una tarea entregable no puede ser finalizada mientras existan otras tareas de ese ítem en progreso. Si un ítem no ha sido finalizado en el último daily scrum de la iteración, debe regresar al listado de producto.
PRIORIDAD	160
PUNTOS DE HISTORIA	6

ID	4-a
NOMBRE	Panel de control de Daily Scrum
DESCRIPCIÓN	Visualización y edición de estados de los ítems a desarrollar en la iteración y sus respectivas tareas (no empezada, en progreso, finalizada), agregar y eliminar tareas al listado y mostrar barra de progreso de ítems de la iteración. La información consignada en el daily scrum debe ser totalmente editable hasta ser confirmada mediante un botón. Una tarea entregable no puede ser finalizada mientras existan otras tareas de ese ítem en progreso. Una tarea no puede ser eliminada si no se encuentra en estado 'pendiente'. Si un ítem no ha sido finalizado en el último daily scrum de la iteración, debe regresar al listado de producto.
PRIORIDAD	150
PUNTOS DE HISTORIA	3

ID	4-b
NOMBRE	Gestión de imprevistos del proyecto
DESCRIPCIÓN	Dividir ítems del listado del producto y crear iteraciones extra. No pueden crearse ítems con la misma prioridad que uno existente. Un ítem vinculado a una iteración no puede ser dividido. Para crear una iteración extra, el proyecto debe haber sido dividido y aun debe haber ítems sin finalizar en el listado del producto.
PRIORIDAD	110
PUNTOS DE HISTORIA	3

Sin embargo, tras la primera división, se decidió realizar una sobre el elemento 4-b con motivo de separar los imprevistos relacionados con el Listado del Producto de los ocurridos con los estimados de iteración, esto con el fin de garantizar la independencia de elementos en el listado.

ID	4-b-a
NOMBRE	Creación de iteraciones extra
DESCRIPCIÓN	Crear iteraciones aunque no fueran contempladas en los estimados de duración del proyecto. Una iteración extra debe tener la misma duración que las anteriores y solo puede ser creada tras finalizados los estimados iniciales de duración del proyecto si aun existen 50tems sin desarrollar en el listado del producto.
PRIORIDAD	105
PUNTOS DE HISTORIA	2
ENTREGABLE	Módulo web que cumpla con las funciones escritas en la descripción.

ID	4-b-b
NOMBRE	División de ítems del listado
DESCRIPCIÓN	División de ítems del listado del producto en ítems más pequeños. No pueden crearse ítems con la misma prioridad que uno existente. Un 50tem vinculado a una iteración no puede ser modificado de ninguna manera.
PRIORIDAD	110
PUNTOS DE HISTORIA	1
ENTREGABLE	Módulo web que cumpla con las funciones escritas en la descripción.

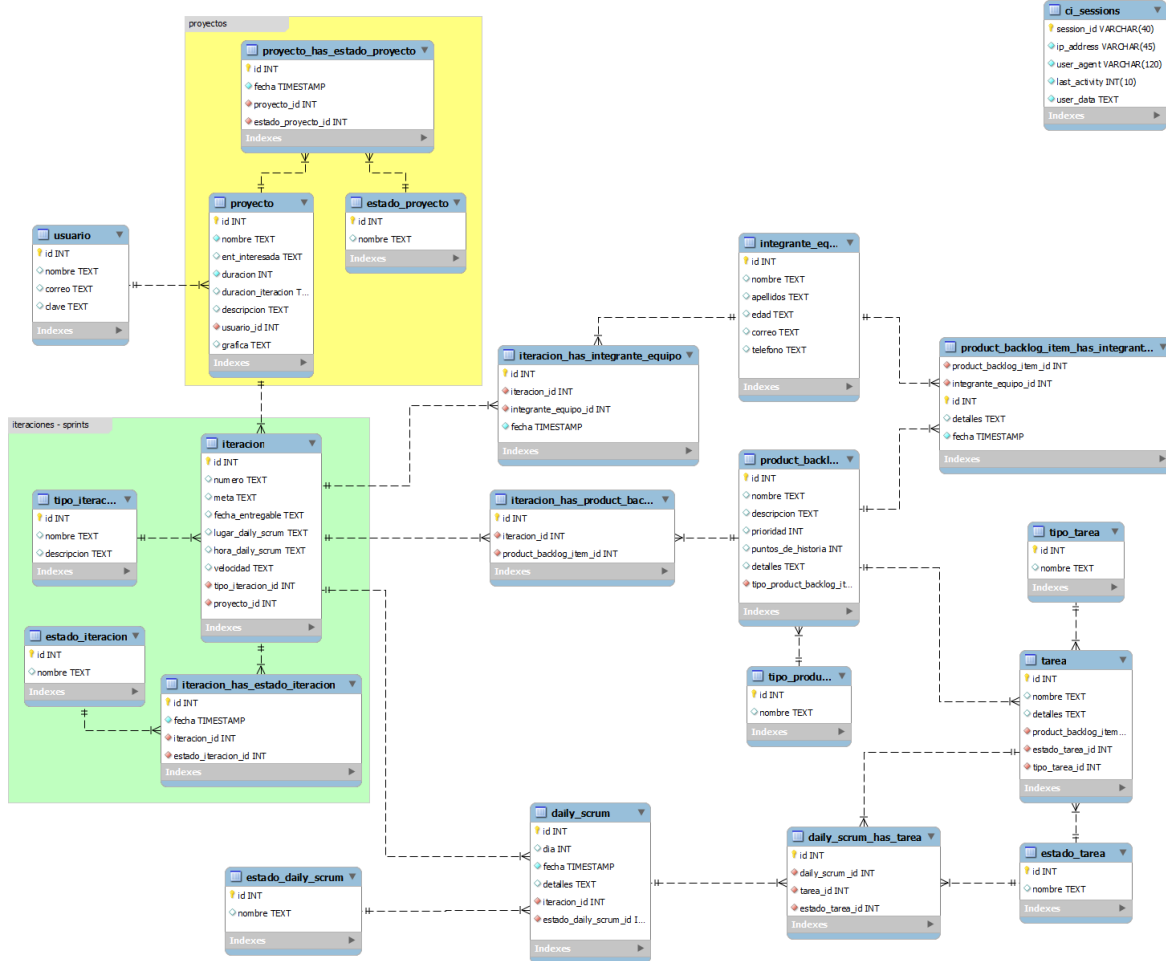
4.2 Desarrollo de la base de datos

En su primera versión, la base de datos respondía correctamente a las necesidades de registro y documentación de los datos de un proyecto, Equipo de Desarrollo y el Listado del Producto. Los elementos del Listado eran enlazados a una iteración 0 correspondiente a la Fase Preliminar de SCRUM, esto con el fin de evitar redundancias en el momento de representar este hecho. Además se evidenció la necesidad de agregar estados a las iteraciones, respondiendo a la posibilidad de iniciar y terminar iteraciones según la fase de SCRUM en que se encontrara el proyecto. La tabla proyecto también presentó necesidad de un estado para poder conocer en qué punto de SCRUM se encontraba un proyecto activo.

Revisando la documentación de SCRUM y analizando las herramientas del estado del arte, se continuaron realizando cambios al esquema original de la base, se cuestionó la necesidad de poder delegar un elemento del Listado del Producto a un integrante del equipo con el fin de establecer un encargado para cada elemento, ya que la metodología defiende el esfuerzo conjunto por encima del individual.

En un movimiento ambicioso y evitando tener que hacer modificaciones en el elemento principal que era el Listado del Producto, se intentó usar la tabla relación entre la iteración y los elementos del Listado como Daily Scrum. Sin embargo, tras un intento fallido se entendió que el Daily Scrum debía ser mas que una relación entre dos tablas, por lo que se decidió agregar en el diseño de la base de datos como una tabla distinta. Se pensó en usar, al igual que la iteración cero, un daily scrum cero donde los elementos del Listado del Producto recién agregados a la iteración se pudieran evidenciar, idea que fue desechada al no demostrar beneficios tangibles.

Se cambió la estructura de las tablas iteración y Daily Scrum para corresponder a la premisa “un elemento del Listado no puede ser modificado de ninguna manera después de haber sido asignado a una iteración”. Como una idea original para esta herramienta, se transformó el atributo “entregable” de un elemento del Listado del Producto en una tarea, dejando como evidencia del progreso de éste al cambio de estados del entregable en el sistema y permitiendo eliminar los estados de la tabla elemento del Listado del Producto.



4.3 Lista de Pendientes de Iteraciones (Completa)

Finalizadas a cabalidad las iteraciones del proyecto, la Lista de Pendientes de las Iteraciones comprende la siguiente información:

Iteración 1		
PBI	1	Fase Inicial de Proyecto
<ul style="list-style-type: none"> - Diseñar y programar vistas. - Validar inicio de sesión. - Crear nuevo usuario. - Verificar al crear usuario que no tenga un correo existente. - Crear nuevo proyecto. - Cargar información de proyecto en variable de sesión. - Validar existencia de miembros del equipo. - Validar existencia de ítems en el listado. 		

- Crear nuevo miembro del equipo.
- Crear nuevo ítem en el listado.
- Verificar al crear un nuevo ítem que no tenga una prioridad existente.
- Editar miembro del equipo.
- Editar ítem del listado.
- Validar campos vacíos y numéricos en formularios.

PBI	2	Fase Preliminar SCRUMs
-----	---	------------------------

- Diseñar y programar vistas.
- Validar precisión de ítems.
- Validar duración de proyectos e iteración.
- Precisar ítem (agregar puntos de historia y entregable).
- Generar tarea entregable.
- Establecer duración de proyecto e iteración.
- Editar duración de proyecto e iteración.
- Editar precisión de ítem.
- Validar campos vacíos y numéricos en formularios.
- Aplicar multiplicadores (día, semana, mes) a duración de proyecto.

Iteración 2		
-------------	--	--

PBI	4-a	Panel de control de daily scrum
-----	-----	---------------------------------

- Diseñar y programar vistas.
- Generar listado de ítems y tareas de iteración.
- Ubicar tareas en el panel según su estado.
- Cambiar estado de tarea.
- Crear nueva tarea.
- Mostrar detalles de ítem.
- Calcular porcentaje de progreso para cada ítem.
- Confirmar daily scrum.
- Comprobar estado de tareas al finalizar un entregable.

Iteración 3		
-------------	--	--

PBI	3	Planeación de iteración
-----	---	-------------------------

- Diseñar y programar vistas.
- Crear nueva iteración.
- Asignar meta, fecha de entregable y lugar y hora del daily scrum.
- Generar listado de miembros del equipo.
- Generar listado de ítems ordenado por prioridad.
- Vincular miembros del equipo a iteración.
- Crear y vincular nuevos miembros.
- Vincular ítems del listado a iteración.
- Verificar que se seleccione al menos un miembro del equipo.
- Verificar que se seleccione al menos un ítem del listado.

			- Verificar prioridad de ítems seleccionados.
PBI	4-b-b	División de ítems del listado	
			- Diseñar y programar vistas. - Validar cantidad de hijas. - Validar campos vacíos. - Validar formularios vacíos. - Dividir 54tem del listado del producto. - Crear 54tems hijos. - Generar tareas entregables hijos. - Verificar que los 54tems hijos no tienen una prioridad ya utilizada.
Iteración 4			
PBI	4-b-a	Creación de iteraciones extra	
			- Diseñar y programar vistas. - Verificar que todas las iteraciones fueron finalizadas. - Verificar que aún hay 54tems pendientes en el listado del producto. - Crear nuevas iteraciones.
PBI	5	Retroalimentación de proyecto	
			- Diseñar y programar vistas. - Calcular puntos de historia terminados por día. - Calcular puntos de historia ideales según estimados iniciales de duración. - Calcular velocidad promedio del equipo. - Calcular velocidad del equipo por iteración. - Generar gráficas. - Calcular cantidades de ítems creados en el proyecto. - Imprimir mensajes de retroalimentación.

5. PRODUCTO

1. Inicio de sesión

En esta ventana se incluye el formulario de inicio de sesión al sistema. Se incluyen los campos de texto de correo electrónico y clave de acceso. El usuario digita la información correspondiente y oprime el botón "Iniciar Sesión". Cuando esté listo; se hace la correspondiente verificación de la base de datos, si los datos son correctos continúa hacia el panel de proyectos, de lo contrario muestra el mensaje de error correspondiente y regresa a esta ventana. Si un usuario no tiene cuenta de ingreso, la opción "Nueva Cuenta" le permite dirigirse a la ventana de creación de cuenta. Si ya existe una sesión activa, esta ventana se ignora y el usuario es dirigido directamente al panel de proyectos.

SCRUM MANAGER

Ya tienes una cuenta? ×

Iniciar Sesión Nueva Cuenta

Iniciar Sesión

Correo

Clave

2. Nueva Cuenta

En esta ventana se incluye el formulario de creación de una nueva cuenta. Se incluyen los campos de texto de nombre de usuario, contraseña y correo electrónico que identificarán al usuario a lo largo de su experiencia con el sistema. El usuario digita la información correspondiente y oprime el

botón “Crear Cuenta” cuando esté listo; se hace la correspondiente verificación de la base de datos, si el correo electrónico no se encuentra ya en uso continúa hacia el panel de proyectos, de lo contrario muestra el mensaje de error correspondiente y regresa a esta ventana. Si el usuario ya tiene una cuenta de ingreso, la opción “Iniciar Sesión” le permite dirigirse a la ventana de Inicio de Sesión.

SCRUM MANAGER

Ya tienes una cuenta?

[Iniciar Sesión](#) [Nueva Cuenta](#)

Nombre de Usuario
prueba

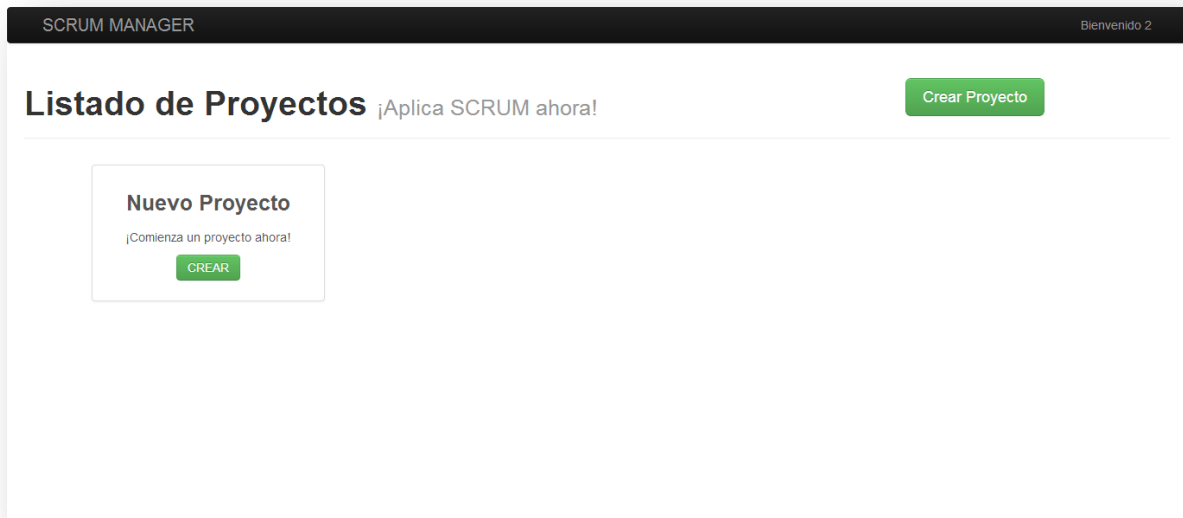
Contraseña
123456789

Correo Electrónico
correo@prueba.com

[Create Account](#)

3. Panel de proyectos

En esta ventana se incluye el listado de proyectos que el usuario ha creado a lo largo de su uso del sistema. Se incluye para cada proyecto el nombre que le identifica y la entidad interesada en éste, además del botón “Ver Proyecto” que permite acceder a la información del proyecto en cuestión. El usuario da click al botón del proyecto del que desea saber mas y se dirige hacia su panel de control. Si su intención es crear un nuevo proyecto, el usuario puede dar click tanto al botón “Crear Proyecto” en la esquina superior derecha como al botón “Crear” del ejemplo de Nuevo Proyecto ubicado al final del listado. Cualquiera de estas opciones llevará al usuario a la ventana de creación de proyectos.



4. Creación de Proyecto

En esta ventana se incluye el formulario de creación de proyectos. Se incluyen los campos de nombre del proyecto, entidad interesada y descripción del proyecto. El usuario digita la información correspondiente y oprime el botón “Confirmar” cuando esté listo; si los campos se encuentran vacíos el formulario no permite la confirmación de los datos, de lo contrario se crea el proyecto y el usuario es dirigido de vuelta al panel de proyectos.

SCRUM MANAGER Bienvenido 2

MIS PROYECTOS

NUEVO PROYECTO

Nombre:
 ✓

Entidad Interesada:
 ✓

Descripción:
 ✓

[CONFIRMAR](#)

5. Menú Principal

El menú está presente a lo largo del sistema desde que se abre un proyecto. El menú se encuentra dividido en los siguientes enlaces:

- Mis Proyectos: Dirige al usuario al Panel de proyectos.
- Información General / Panel de Control: Dirige al usuario al panel de control del proyecto activo.
- Descripción: Dirige al usuario a la ventana de descripción del proyecto.
- Equipo de trabajo: Dirige al usuario a la ventana de equipo de trabajo.
- Historias de Usuario: Dirige al usuario a la ventana de Listado del producto.
- Iteraciones: Dirige al usuario a la ventana de Listado de Iteraciones.
- Cerrar Sesión: Finaliza la sesión activa y regresa a la ventana de inicio de sesión.



6. Panel de Scrum Master

El panel de control es la columna vertebral del sistema, funciona como ventana principal de un proyecto y su contenido varía según la etapa de la metodología SCRUM en la que el proyecto se encuentre. Los siguientes son los posibles estados gráficos del Panel de Control:

6.1. Fase inicial del proyecto

Cuando un proyecto está recién creado, el panel de control entra directamente a esta fase. La ventana es formada por cuatro botones alargados que representan las dos primeras fases de un proyecto SCRUM y un quinto botón que permite ir a la siguiente fase del proyecto. Los 4 botones iniciales cambian según la interacción del usuario con el sistema de la siguiente manera:

- *Crear grupo de trabajo/pendiente:*
Es de color anaranjado, dirige al usuario a la ventana de “Crear miembro del equipo”.
- *Crear grupo de trabajo/finalizado:*
Su texto cambia a “Modificar grupo de trabajo” y el color del botón a verde. Dirige al usuario a la ventana de “Editar miembros del equipo”.

- *Crear pila de producto/pendiente:*
Es de color anaranjado, dirige al usuario a la ventana de “Crear ítem del Listado de Producto”.
- *Crear pila de producto /finalizado:*
Su texto cambia a “Administrar pila del producto” y el color del botón a verde. Dirige al usuario a la ventana de “Listado del Producto”.
- *Precisar ítems de la pila de producto/pendiente:*
Dirige al usuario a la ventana de “Listado del Producto”.
- *Precisar ítems de la pila de producto/finalizado:*
Su texto cambia a “Precisar ítems de la pila de producto” y el color del botón a verde. Dirige al usuario a la ventana de “Listado del Producto”.
- *Seleccionar duración del proyecto/pendiente:*
Dirige al usuario a la ventana de “Duración del proyecto”.
- *Seleccionar duración del proyecto/finalizado:*
Su texto cambia a “Editar duración del proyecto” y el color del botón a verde. Dirige al usuario a la ventana de “Duración del proyecto”.

Los estados de cada uno de estos botones dependen de la acción subsecuente del usuario sobre cada una de las ventanas a las que dirigen. El quinto botón de color rojo y etiqueta “SCRUM!” permanece inhabilitado, hasta que todos los demás botones se encuentren en su estado finalizado. Cuando el usuario da click a este botón, el proyecto avanza a una siguiente fase y la ventana de panel de control evoluciona junto con él.



7. Crear Miembro del Equipo

SCRUM MANAGER Bienvenido 2

MIS PROYECTOS

MI PROYECTO

- Panel de SCRUM Master
- Información del proyecto
- Equipo de Trabajo
- Listado del Producto
- Iteraciones

CONFIGURACIÓN

CERRAR SESIÓN

Mi Proyecto

Nuevo integrante del grupo de trabajo

Nombres:
 ✓

Apellidos:
 ✓

Edad:
 ✓

Correo electrónico:
 ✓

Teléfono:
 ✓

En esta ventana se incluye el formulario de creación de miembros del equipo de desarrollo. Se incluyen los campos de texto de nombre, apellidos, edad, correo electrónico y teléfono. El usuario digita la información correspondiente y oprime el botón "Agregar" cuando esté listo; si alguno de los campos se encuentra vacío, el formulario no debe

8. Crear pila del producto

SCRUM MANAGER Bienvenido 2

MIS PROYECTOS

MI PROYECTO

- Información del proyecto
- Equipo de Trabajo
- Listado del Producto
- Iteraciones

CONFIGURACIÓN

CERRAR SESIÓN

NUEVO ITEM DEL LISTADO DE PRODUCTO CONFIRMAR

Nombre: ✓

Prioridad: ✓

Descripción: ✓

En esta ventana se agregaron los primeros elementos que se conocen de un PBI (Elemento del listado del producto) con el fin de mantener el orden de la metodología SCRUM.

La prioridad se maneja en rangos indefinidos, siendo el mismo equipo de trabajo el que los define. El software obliga al scrum master a considerar las prioridades de mayor a menor, siendo la mayor la más prioritaria. Dos elementos no podrán tener la misma prioridad.

9. Precisar elemento del listado

SCRUM MANAGER Bienvenido 1

MIS PROYECTOS

MI PROYECTO

- Panel de SCRUM Master
- Información del proyecto
- Equipo de Trabajo
- Listado del Producto
- Iteraciones

CONFIGURACIÓN

CERRAR SESIÓN

PRECISAR ELEMENTO DEL LISTADO

Nombre:

Prioridad:

Puntos de historia:

Entregable:

Descripción:

Confirmar

Luego de haber realizado una reunión para saber lo que se realizará para cada elemento del listado del producto y también su dificultad la cual está representada en puntos de historia, se procede a editar estos datos en cada elemento a la que no se le han agregado. Este paso es indispensable para poder continuar con el proceso.

También se asigna la tarea entregable que se tendrá en cuenta para saber si el elemento fue terminado o continua pendiente.

10. Inicio de iteración.

The screenshot displays the 'SCRUM MANAGER' interface. On the left, a sidebar menu includes 'MIS PROYECTOS', 'MI PROYECTO', 'Panel de SCRUM Master', 'Información del proyecto', 'Equipo de Trabajo', 'Listado del Producto', 'Iteraciones', 'CONFIGURACIÓN', and 'CERRAR SESIÓN'. The main content area is titled 'Mi Proyecto' and 'ITERACIÓN 1'. It contains several configuration fields, each with a green checkmark indicating it is set:

- Meta:** Ofrecer seguridad al cliente
- Fecha DEMO de iteración:** 25 de Octubre de 2013
- Lugar Daily Scrum:** En la oficina
- Hora Daily Scrum:** 3:00 pm

A blue 'CONFIRMAR' button is located at the bottom of the configuration area.

Luego al iniciar el SCRUM se procede a crear la primera iteración. Los datos que se agregarán corresponden a una meta, una fecha de DEMO donde se representa el día en el que se demuestra la finalización. También se agregan el lugar y la hora del Daily SCRUM los cuales se deben mantener durante toda la iteración.

11. Selección de equipo para la iteración.

SCRUM MANAGER Bienvenido 1

MIS PROYECTOS

MI PROYECTO

- Panel de SCRUM Master
- Información del proyecto
- Equipo de Trabajo
- Listado del Producto
- Iteraciones

CONFIGURACIÓN

CERRAR SESIÓN

Selecciona tu equipo de trabajo para la iteración 1

Provenientes de iteraciones anteriores:

	Nombres	Apellidos	Edad	Correo	Telefono	
<input type="checkbox"/>	Heiner Wbeimar	Maldonado	23	heinerangarita@gmail.com	3173745559	✎ ✕

[Nuevo Usuario](#)

Una persona creada en este punto quedará inscrita automáticamente como parte del equipo de esta iteración

[Confirmar equipo](#)

Luego se seleccionará el equipo que estará disponible para la iteración, lo que nos lleva a pensar en agregar integrantes que no han estado en otras iteraciones. Si es el caso, pueden agregarse integrantes en este punto pero quedaran ligados inmediatamente a la iteración. Se debe seleccionar al menos un integrante para poder continuar.

12. Selección de ítems de la lista del producto.

SCRUM MANAGER Bienvenido 1

MIS PROYECTOS

MI PROYECTO

- Panel de SCRUM Master
- Información del proyecto
- Equipo de Trabajo
- Listado del Producto
- Iteraciones

CONFIGURACIÓN

CERRAR SESIÓN

Selecciona los items de la pila de producto a desarrollar esta iteración

	Prioridad	Titulo	Puntos de historia	Descripcion	Entregable	
<input type="checkbox"/>	150	Crear modulo de planeación	10	Elemento que ayudará a realizar correctamente la documentación de los temas hablados en cada reunion	modulo web que cumpla con los requerimientos	✎ ✕
<input type="checkbox"/>	100	Crear login y password	25	Crear un medio de seguridad para que el usuario mantenga su información solo enlazada a él	algo muy importante!	✎ ✕

[Confirmar Items](#)

Igualmente en la siguiente ventana se deben seleccionar los ítems del Listado del Producto que serán usados en la iteración. Se debe tener en cuenta que en el momento en el que se insertan a la iteración dejan de ser editables, eliminables, divisibles entre otras, transformándolas en “intocables”.

No se puede continuar en el proceso si no se ha seleccionado por lo menos un ítem.

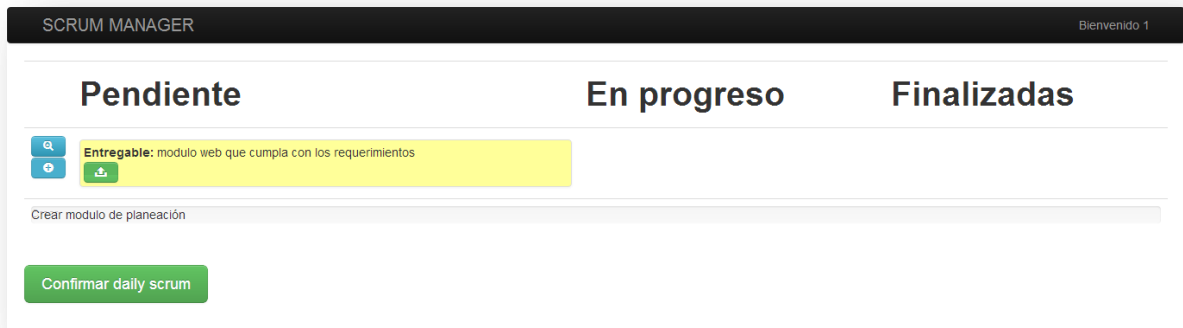
13. Descripción básica de la iteración.

The screenshot displays the SCRUM MANAGER interface. At the top, there is a header with 'SCRUM MANAGER' on the left and 'Bienvenido 1' on the right. A sidebar on the left contains navigation options: 'MIS PROYECTOS', 'MI PROYECTO', 'Panel de SCRUM Master' (highlighted), 'Información del proyecto', 'Equipo de Trabajo', 'Listado del Producto', 'Iteraciones', 'CONFIGURACIÓN', and 'CERRAR SESIÓN'. The main content area is titled 'Mi Proyecto' and features a red 'DAILY SCRUM!' button. Below the title, it shows 'ITERACIÓN 1 / 2' and 'Faltan 10 daily scrums'. A table lists project details: Meta (Ofrecer seguridad al cliente), Fecha DEMO de iteración (25 de Octubre de 2013), and Lugar y Hora Daily SCRUM (En la oficina / 3:00 pm). Under 'Equipo de trabajo', the name 'Heiner Wbeimar Maldonado' is listed. The 'Items a desarrollar' section shows 'Crear modulo de planeación' with a status of 'Pendiente (0%)'.

En esta ventana se muestran los datos relevantes con respecto al estado actual de la iteración; se pueden ver quienes hacen parte del equipo de trabajo, cual es la meta, lugar y hora del Daily Scrum, los ítems que se encuentran en la iteración además de su estado actual.

En este punto ya se podrá realizar cada Daily Scrum.

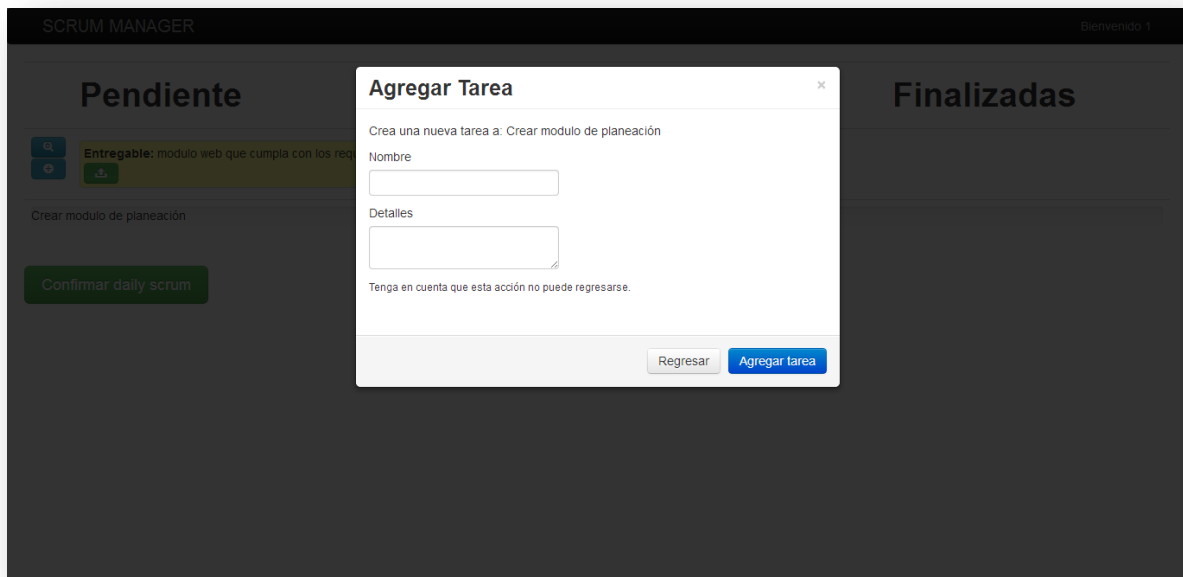
14. Daily SCRUM



Esta ventana es la más compleja de todo el software siendo la que más movimiento va a tener en el transcurso de toda la iteración.

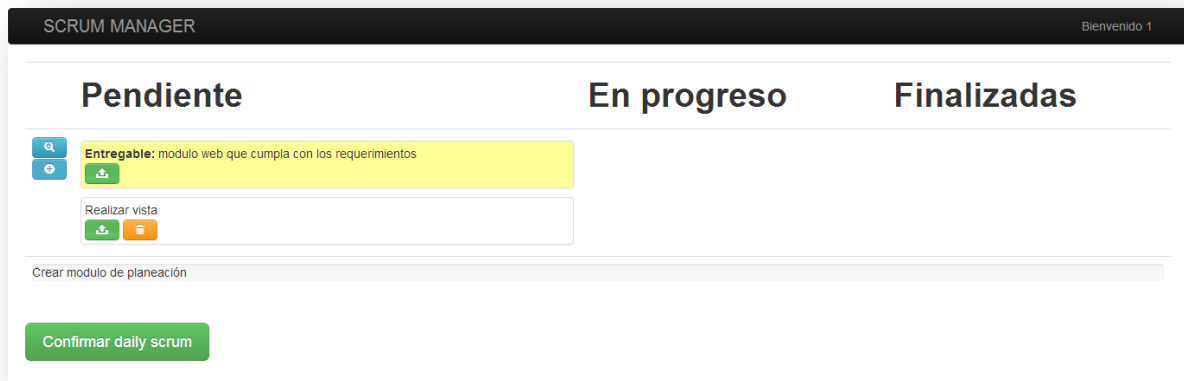
Las acciones posibles que se pueden ejecutar son los siguientes:

14.1. Agregar Tareas:



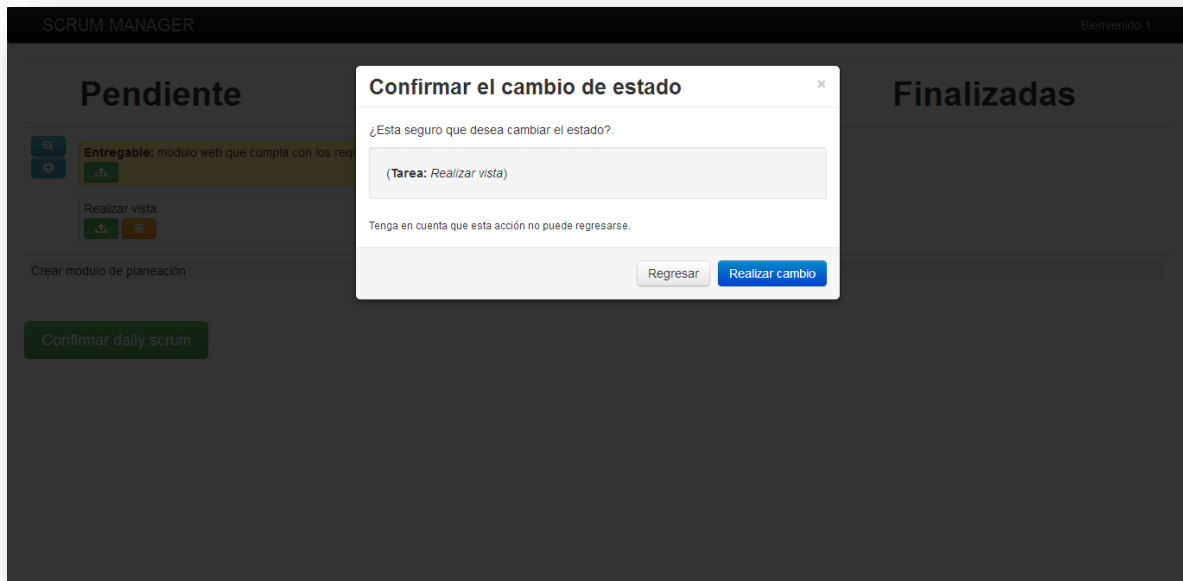
El cambio está siempre presente en las iteraciones por lo que es muy difícil saber las tareas que se tendrán que realizar en cada ítem del producto. La posibilidad de solo mover los ítems del Listado del Producto por el tablero está contemplada al poderse mover el entregable (El elemento amarillo), pero también es posible crear tareas para poder llevar una documentación un poco más precisa.

14.2. Eliminar una tarea:



Siempre y cuando una tarea se encuentre pendiente, es posible eliminarla. Esta función se dejó disponible para corregir errores o eliminar elementos innecesarios en la documentación.

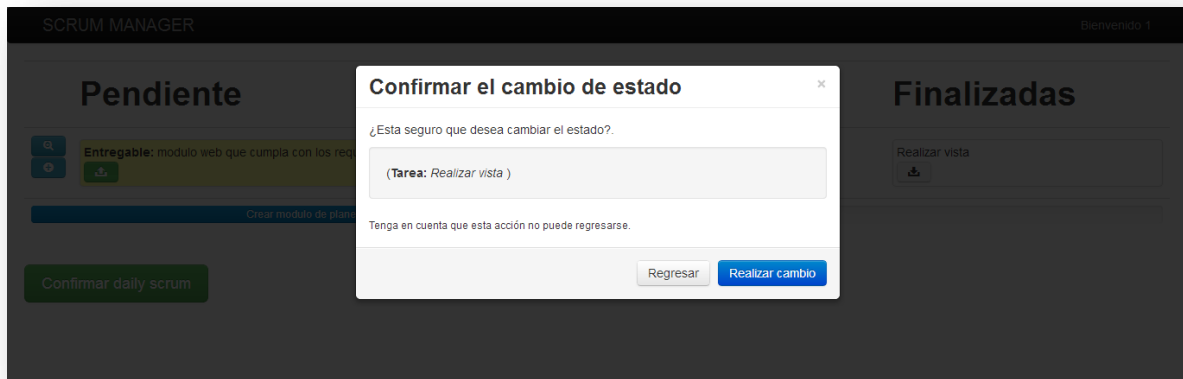
14.3. Aumentar el estado de una tarea:



Al hacer click en el botón verde se mostrará una ventana para confirmar el cambio de estado de la tarea; al hacerlo la tarea pasa al estado superior más cercano indicando su avance. Si una tarea es marcada como finalizada, el ítem del Listado del Producto avanzará en porcentaje según la cantidad de tareas que han sido realizadas pero esto es solo un indicador visual, no genera un real movimiento en la gráfica de rendimiento.

Una tarea entregable no puede ser marcada como finalizada si existe al menos una tarea del mismo ítem que no ha sido finalizada.

14.4. Disminuir el estado de una tarea



Al hacer click en el botón plateado se mostrará una ventana para confirmar el cambio de estado de la tarea; al hacerlo la tarea pasa al estado inferior más cercano indicando su retroceso, el ítem cambiará en porcentaje según la cantidad de tareas que han sido realizadas pero esto es solo un indicador visual, no genera un real movimiento en la gráfica de rendimiento.

La opción se ha mantenido habilitada para poder corregir errores posibles en el transcurso del uso de la herramienta.

Un ítem no puede ser disminuido de estado si el entregable se encuentra en estado finalizado.

14.5. Confirmar Daily SCRUM:

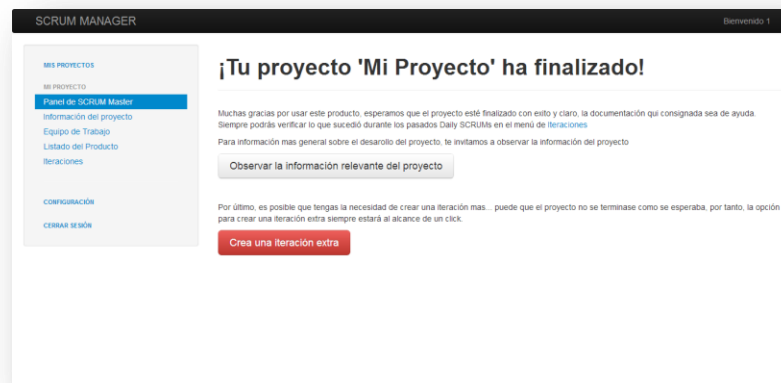
Se da por hecho que el Daily Scrum ha sido ingresado con éxito y se procederá a ingresar el siguiente en una próxima oportunidad.

15. Finalizar proyecto

Al terminar todos los Daily SCRUM y todas las iteraciones, el proyecto está listo para ser finalizado. El botón de “siguiente iteración” se transformará en “FINALIZA EL PROYECTO” y al hacerlo se recargará el panel de SCRUM Master



Inmediatamente luego de finalizar el proyecto aparecerá la ventana de Información del proyecto (descrita en el siguiente numeral). Si se llega a entrar de nuevo al Panel de SCRUM Master, la opción de crear una nueva iteración se activa permitiendo continuar con el proyecto incluso si ya terminó el tiempo disponible.



Esta opción permite iniciar una iteración como si fuera parte del proyecto.

SCRUM MANAGER
Bienvenido 1

MIS PROYECTOS

MI PROYECTO

Panel de SCRUM Master

Información del proyecto

Equipo de Trabajo

Listado del Producto

Iteraciones

CONFIGURACIÓN

CERRAR SESIÓN

Mi Proyecto

Duración	20 días
Duración de iteración	10 días
Entidad interesada	Nosotros
Descripción	Proyecto de prueba para probar el sistema

Porcentaje de proyecto terminado:

0 %

¡Tu equipo debe esforzarse mas! su trabajo no es acorde a sus estimados iniciales, tienes 1 días de retraso

Trabaja en aumentar la velocidad de tu equipo, tal vez existen problemas de ánimo
Las iteraciones extra siempre son una opción para ti, pero tal vez no para tu cliente

Velocidad promedio del equipo:

0 puntos de historia / día
0 puntos de historia / iteracion

Información de PBIs:

Total:
2

En el listado del producto:
1

En progreso:
1

Finalizados:
0

Divididos:
0 (No entra en el total)

Hijos:
0

En esta ventana se puede evidenciar el avance del proyecto siendo una ayuda visual para el SCRUM Master. Puede visualizarse en cualquier momento del proyecto para conocer en qué estado se encuentra el mismo.

En este mismo espacio se observa la Gráfica de rendimiento pero solo puede verse cuando algún ítem ha sido finalizado.

- MIS PROYECTOS
- MI PROYECTO
 - Panel de SCRUM Master
 - Información del proyecto**
 - Equipo de Trabajo
 - Listado del Producto
 - Iteraciones
- CONFIGURACIÓN
- CERRAR SESIÓN

Mi Proyecto

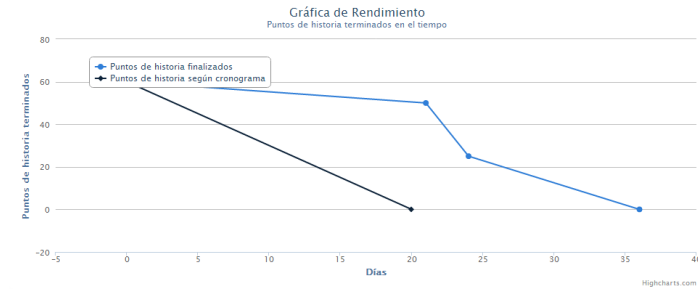
Duración	20 días
Duración de iteración	9 días
Entidad interesada	Nosotros
Descripción	Proyecto de prueba para el software

Porcentaje de proyecto terminado:



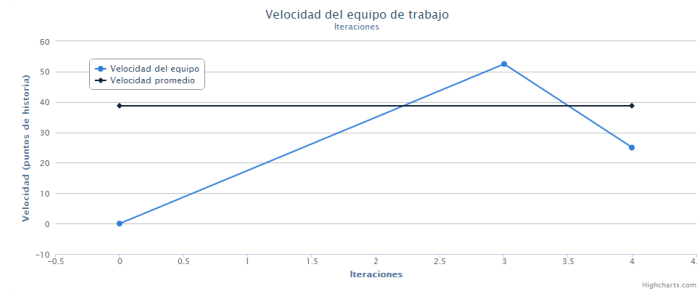
¡Tu equipo debe esforzarse mas! su trabajo no es acorde a sus estimados iniciales, tienes **16 días de retraso**

Trabaja en aumentar la velocidad de tu equipo, tal vez existen problemas de ánimo. Las iteraciones extra siempre son una opción para ti, pero tal vez no para tu cliente



Velocidad promedio del equipo:

4.30555555555556 puntos de historia / día
38.75 puntos de historia / iteración



Información de PBIs:

Has fallado en terminar 2 PBI durante tus iteraciones, apóyate en la velocidad ponderada de tu equipo y su estado de ánimo para tomar mejores decisiones

Has creado 1 PBI que no se tuvieron en cuenta antes de iniciar las iteraciones. Controla esto con mejores prácticas en las reuniones preliminares.

- Total:**
3
- En el listado del producto:**
0
- En progreso:**
0
- Finalizados:**
3
- Divididos:**
0 (No entra en el total)
- Hijos:**
0

17. Equipo de trabajo

SCRUM MANAGER Bienvenido 1

MIS PROYECTOS

- MI PROYECTO
- Panel de SCRUM Master
- Información del proyecto
- Equipo de Trabajo**
- Listado del Producto
- Iteraciones

CONFIGURACIÓN

CERRAR SESIÓN

Nombres	Apellidos	Edad	Correo	Telefono	
Heiner Wbelmar	Maldonado	23	heinerangarita@gmail.com	3173745559	✎ ✕
Luis Gabriel	Rodriguez	22	luisgtoscano@gmail.com	3187652737	✎ ✕

En esta ventana se pueden observar los integrantes del equipo relacionados a la iteración actual. Si no se ha iniciado ninguna iteración, en esta ventana se permiten agregar integrantes al proyecto con el fin de poder agregarlos en las iteraciones.

18. Listado del producto

SCRUM MANAGER Bienvenido 1

Crear PBI

Listado del Producto

Items que aun no han sido asignados a una iteración

Prioridad	Titulo	Puntos de historia	Encargado	Descripción	Entregable	
100	Crear login y password	25	Aun no tiene encargado	Crear un medio de seguridad para que el usuario mantenga su información solo enlazada a él	algo muy importante!	✎ ✕

En esta ventana se podrá agregar, dividir y modificar los elementos del listado del producto, siendo posible modificarlos en cualquier momento del proyecto. Los elementos que no se encuentren en una iteración aparecerán en esta ventana para poder ser modificados.

18.1. División de elemento del listado del producto

The screenshot shows the 'SCRUM MANAGER' interface with a 'Bienvenido 1' greeting in the top right. On the left, a sidebar contains navigation links: 'MIS PROYECTOS', 'MI PROYECTO', 'Panel de SCRUM Master', 'Información del proyecto', 'Equipo de Trabajo', 'Listado del Producto', 'Iteraciones', 'CONFIGURACIÓN', and 'CERRAR SESIÓN'. The main content area is titled 'DIVIDIR HISTORIA DE USUARIO'. It features a form with the following fields: 'Nombre' (text input with 'Crear login y password'), 'Prioridad' (text input with '100'), 'Puntos de historia' (text input with '25' and 'días' label), and 'Entregable' (text input with 'algo muy importante!'). A large text area for 'Descripción' contains the text 'Crear un medio de seguridad para que el usuario mantenga su información solo enlazada a él'. At the bottom, a summary bar shows 'Deseo dividir esta historia en 2 partes' followed by a blue 'Confirmar' button.

Es tan simple como indicar el número de elementos en el que se dividirá para luego ir a un formulario donde se crearan los nuevos elementos.

SCRUM MANAGER Bienvenido 1

MIS PROYECTOS

MI PROYECTO

- Panel de SCRUM Master
- Información del proyecto
- Equipo de Trabajo
- Listado del Producto
- Iteraciones

CONFIGURACIÓN

CERRAR SESIÓN

DIVIDIR ELEMENTO "Crear login y password"

Historia hija 1

Nombre	<input type="text"/>	Descripción	<div style="border: 1px solid #ccc; height: 100px;"></div>
Prioridad	<input type="text"/>		
Puntos de historia	<input type="text"/>		
Entregable	<input type="text"/>		

Historia hija 2

Nombre	<input type="text"/>	Descripción	<div style="border: 1px solid #ccc; height: 100px;"></div>
Prioridad	<input type="text"/>		
Puntos de historia	<input type="text"/>		
Entregable	<input type="text"/>		

Se deberá diligenciar completamente el formulario para poder evidenciar realmente el cambio, sino, la acción se perderá y se deberá indicar nuevamente que se desea dividir el ítem.

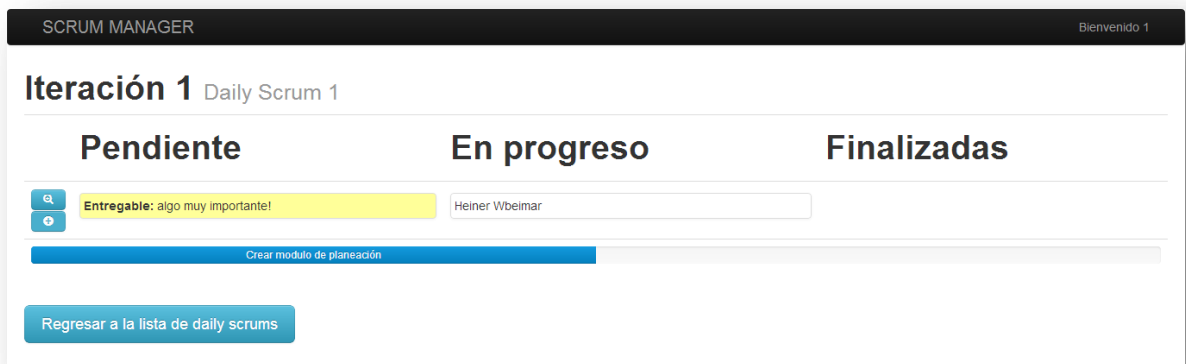
19. Iteraciones

El menú de iteraciones nos permite saber cuántos Daily SCRUM se han realizado hasta el momento y saber cómo se desarrolló cada uno con el fin de llevar un control mucho más exacto sobre cada día. Es posible también editar un Daily SCRUM si es el último y no se ha confirmado.

76



Al seleccionar un día desarrollado aparecerá el panel de control de Daily SCRUM mostrando como quedaron las tareas que fueron registradas en el momento.



6. PRUEBAS

Al iniciar las pruebas se establecieron primero los procedimientos básicos para poder realizarlas, lo que llevó a dividir las pruebas en dos pruebas.

Pruebas de enlace:

Estas pruebas corresponden a la navegabilidad del sitio, garantizando que todos los links de todas las páginas lleven a algún lugar y sea posible regresar de posibles errores.

Pruebas de formulario:

Estas pruebas consisten en agregar datos erróneos a los formularios y ver su comportamiento.

Para la primera prueba se utilizó la página <http://validator.w3.org/checklink> siendo una de las más recomendadas por la comunidad.

Al ejecutarse el test, algunos links resultaron estar rotos, en su mayoría herramientas externas al programa que se habían implementado con el fin de obtener rápidamente los resultados esperados. Para evitar este problema se decidió agregar todos los archivos directamente al proyecto y evitar tener problemas por terceros.

Los links internos fueron también analizados y se encontraban en correcto enlace con las demás funcionalidades del sistema.

Las pruebas de formulario se realizaron manualmente, siendo posible la aparición de errores no probados. Cada formulario de la aplicación fue probado con el siguiente tipo de caracteres:

1. Caracteres especiales
2. Letras acentuadas
3. Números
4. Espacios vacíos

En general los formularios fueron adecuados para evitar el envío de datos incorrectos y las pruebas fueron acordes a este hecho.

Fueron probadas también las páginas de control entre un formulario y su acción en la base de datos para evitar errores; todos respondieron exitosamente al fallo y rechazaron cualquier manejo de las funciones sin previa verificación en los formularios correspondientes.

7. CONCLUSIONES Y RECOMENDACIONES

Finalizado el proyecto y tras haber implementado cada uno de los eventos descritos por la metodología SCRUM, se ha desarrollado una herramienta software que cumple a cabalidad con los objetivos propuestos dentro del tiempo estimado para el proyecto. Basándonos en investigación y documentación previa, análisis del estado del arte y experiencia adquirida a través de la implementación y desarrollo de este proyecto, podemos realizar con certeza las siguientes afirmaciones respecto al uso de la metodología SCRUM para proyectos de software:

7.1 Conclusiones

- El uso de la metodología resulta plenamente satisfactorio para el Equipo de Desarrollo. Los desarrollos incrementales representan una motivación constante para la realización de buenas prácticas y el entendimiento con el cliente. La naturaleza auto-organizada y multidisciplinaria del Equipo de Desarrollo desencadena en éste un ambiente de unión y responsabilidad que genera resultados superiores a experiencias de equipos administrados en organigrama vertical.
- La transparencia y naturaleza holística de la metodología permiten que tanto el cliente como el Equipo SCRUM verifiquen el estado del proyecto en cualquier punto del desarrollo, esta característica garantiza la identificación oportuna de errores y necesidad de cambio a lo largo del proyecto, y permite la satisfacción constante de todas las partes involucradas.
- El éxito de la metodología SCRUM depende en gran medida de la correcta realización de los eventos descritos en ésta. Aunque pudiera resultar tedioso seguirlos al pie de la letra, solo de esta manera se obtendrán los resultados esperados; razón por la cual el rol del Scrum Master asume la mayor responsabilidad del proceso y es el centro del desarrollo de esta aplicación.
- La utilización del sistema de estimación relativa por puntos de historia presenta ventajas tangibles en planeación preliminar de proyectos sobre los métodos tradicionales de estimación en unidades de tiempo. La estimación relativa permite optimizar la capacidad de esfuerzo del Equipo de Desarrollo y realizar predicciones mas precisas y confiables, unificando además el entendimiento del Equipo sobre el incremento a desarrollar.

- La metodología de programación en parejas tiene resultados óptimos en la detección oportuna de errores de desarrollo, estandarización de código y cohesión de equipo.
- El uso correcto de SCRUM no garantiza el resultado óptimo de un proyecto software. Los proyectos software son complejos, cambiantes y poco predecibles; SCRUM garantiza la transparencia del proyecto para todos los involucrados y fortalece las comunicaciones del equipo y el cliente, lo cual minimiza riesgos y da flexibilidad al cambio y manejo de errores. Sin embargo, debe complementarse con prácticas de desarrollo, arquitectura, diseño, trazabilidad y manejo de versiones, entre otras, para garantizar tener resultados óptimos.
- El uso de PHP como lenguaje base en el desarrollo del proyecto cumple con su objetivo de realizar una codificación rápida y acorde al cronograma establecido. Sin embargo, presenta dificultades de organización de código y una estructura de control de flujo compleja y poco comprensible; obstáculos que se vieron disminuidos gracias a la utilización del framework codeigniter, que optimiza a gran escala el desarrollo web con este lenguaje.

7.2 Recomendaciones

- SCRUM es una metodología intrínsecamente humana y de las interacciones sociales del equipo dependen resultados óptimos. Por esta razón, se recomienda evitar que los soportes tecnológicos y herramientas software se involucren directamente en la realización de los eventos descritos en la metodología, y se destinen por otra parte a optimizar procesos de documentación y retroalimentación posteriores.
- La asignación de tareas a cada elemento del Listado del Producto representa una guía muy valiosa para el Equipo de Desarrollo en el entendimiento de la definición de “Terminado” y la transparencia del proyecto. Sin embargo, se recomienda evitar que el cumplimiento de tareas sea tomado en cuenta para calcular el rendimiento del proyecto a nivel de iteración, para así evitar caer en prácticas micro-administrativas alejadas del objetivo de la metodología.

- La metodología SCRUM rechaza los fallos o méritos individuales en la entrega de incrementos. Se recomienda evitar la asignación de miembros del Equipo de Desarrollo como encargados de elementos del Listado del Producto, para de esta manera asumir los logros o incumplimientos del proyecto del equipo como un todo.

- Para versiones posteriores del software que quisieran abordar más allá del prototipo recomendaríamos ampliar la flexibilidad del tablero de Daily Scrum y del Listado del Producto, permitiendo al usuario crear sus propios estados para el tablero y sus propios atributos de elementos del Listado, permitiendo así al usuario personalizar su proceso SCRUM como dicta la metodología.

8. GLOSARIO

En esta sección se incluyen algunos de los términos más usados dentro de la metodología SCRUM y otros que pudieran presentar duda o confusión. Se espera que sea un instrumento de referencia rápida, de utilidad para quienes no son conocedores de la metodología.

Agilidad

Ciclo de desarrollo caracterizado por la ejecución en paralelo y de forma concurrente de las diferentes fases de ejecución: análisis, diseño, producción, pruebas e integración, permitiendo la construcción o modificación del producto de forma continua a través de la entrega continuada de incrementos discretos funcionalmente operativos y válidos para su integración y funcionamiento en el sistema en desarrollo.

Gráfico de avance (*Burn down*)

El gráfico de avance o “burndown” es el gráfico que actualiza el equipo en las reuniones de seguimiento del sprint, para monitorizar el ritmo de avance, y detectar de forma temprana posibles desviaciones sobre la previsión que pudieran comprometer la entrega al final de sprint.

Historia de usuario

Descripción de una funcionalidad que debe incorporar un sistema de software, y cuya implementación aporta valor al cliente.

La estructura de una historia de usuario está formada por:

1. Nombre breve y descriptivo.
2. Descripción de la funcionalidad en forma de diálogo o monólogo del usuario describiendo la funcionalidad que desea realizar.
3. Criterio de validación y verificación que determinará para considerar terminado y aceptable por el cliente el desarrollo de la funcionalidad descrita.

Y adicionalmente por la información que resulte necesaria por el modelo de implementación: Prioridad, Riesgo, Tamaño, etc.

Manifiesto ágil

En marzo de 2001, 17 críticos de los modelos de mejora basados en procesos, convocados por Kent Beck, que había publicado un par de años antes el libro “Extreme Programming Explained” en el que exponía una nueva metodología denominada Extreme Programming, se reunieron en Salt Lake City para discutir sobre el desarrollo de software.

En la reunión se acuñó el término “Métodos Ágiles” para definir a los que estaban surgiendo como alternativa a las metodologías formales, (CMM-SW, PMI, SPICE) a las que consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas, previas al desarrollo.

Listado del producto (*Product backlog*)

En inglés product backlog Lista ágil de los requisitos del cliente o requisitos del sistema:

- Simples: Expresados de forma breve, con una sentencia para cada uno, habitualmente con formato de historia de usuario o similar.
- Estimados: Está estimado el esfuerzo de construcción de cada requisito.
- Priorizados: Ordenados según la importancia para el cliente o responsable de la lista.

Punto de historia (*Story Point*)

Unidad de trabajo empleada habitualmente en Extreme Programming, definida por la cantidad de trabajo realizada en un día de trabajo ideal (tiempo ideal).

Propietario de producto (*Product owner*)

Término habitual para designar el rol de “responsable de producto” o “product manager” en equipos ágiles.

Desempeñado por una persona (una sólo) conocedora del entorno del negocio del cliente y de la visión del producto, y que representa a todos los interesados en el producto final.

Es el responsable de la pila del producto, y su responsabilidad es determinar cuál es el resultado posible de mayor valor para los usuarios o clientes. Se responsabiliza de las decisiones sobre fechas y funcionalidades de las diferentes versiones del producto, y conoce las posibilidades y plan

de inversión, así como el retorno esperado de la inversión del proyecto. En desarrollos internos puede ser el product manager o responsable de marketing quien asuma este rol. En desarrollos para clientes externos lo más aconsejable es que sea el responsable del proceso de adquisición del cliente.

Scrum Master

En las implantaciones de Scrum estándar el rol responsable del funcionamiento de scrum en la organización. Es un facilitador, que ajusta su intervención en función de la madurez del equipo y de la organización para lograr el funcionamiento del marco académico de scrum, actuando desde formador y entrenador del modelo, a supervisor y facilitador para la realización de las prácticas de Scrum y resolución de problemas e impedimentos.

Sprint (Iteración)

Ciclo de tiempo en el que se desarrolla cada incremento iterativo del producto.

En función de las características del proyecto y el criterio del equipo, lo habitual es realizar sprints de duración no inferior a una semana ni mayor de un mes. Cada sprint produce un incremento.

Tarea

Unidad de trabajo gestionada por el equipo. Una tarea tiene asignada una persona para su realización, y es recomendable que el esfuerzo estimado parara llevarla a cabo sea como máximo el equivalente al realizable en una jornada de trabajo. Los equipos de trabajo ágiles descomponen las historias de usuario se descomponen en tareas para gestionar y seguir el avance de su ejecución.

Programación en pareja

La programación en pareja (pair programming en inglés) es una técnica empleada en el desarrollo ágil de software, consistente en trabajar en el mismo equipo dos programadores de forma conjunta. Uno de ellos (el conductor) escribe el código, mientras que el otro (observador) lo supervisa. Ambos programadores van alternando estos roles.

9. BIBLIOGRAFÍA

- Chin, G. (2004). *Agile Project Management: How to Succeed in the Face of Changing Project Requirements*. AMACOM.
- Cockburn, A. (2000). *Agile Software Development*. Highsmith Series.
- Ellislab. (s.f.). *Página oficial de Codeigniter*. Obtenido de <http://ellislab.com/codeigniter>
- Foundation, P. (s.f.). *PHP*. Obtenido de <http://www.php.net/index.php>
- Foundation, M. (s.f.). *Página oficial de MySQL*. Obtenido de <http://www.mysql.com/>
- Manifiesto for Agile Software Development. (s.f.).
- Network, M. D. (2013). *Artifacts, Work ítems and Workflow, User Story, Defining a User Story*.
- Palacio, J. (s.f.). *Flexibilidad con Scrum: Principios de diseño e implantación de campos de Scrum*.
- Schwaber, K. (s.f.). *Advanced Development Methods. SCRUM Development Process*.
- Sutherland, J. (2004). *Agile Development: Lessons learned from the first Scrum*.
- Sutherland, J., & Schwaber, K. (2011). *The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game*.
- Takeuchi, H., & Nonaka, I. (1986). *The New New Product Development Game*. *Harvard Business Review*.
- w3c. (s.f.). *Hojas de estilo*. Obtenido de <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
- w3techs. (s.f.). *Diagrama estadístico de uso de bibliotecas basadas en javascript*. Obtenido de http://w3techs.com/technologies/overview/javascript_library/all
- Wake, B. (2003). *INVEST in Good Stories, and SMART Tasks*.
- Wave™, T. F. (2009). *Enterprise Database Management Systems*. *The Forrester Wave™*, 7.