**CAPITULO 4- METODOLOGIA DE DESARROLLO.**

La metodología de desarrollo elegida para el presente proyecto es SCRUM. Este es un marco de trabajo para la gestión y [desarrollo de software](http://es.wikipedia.org/wiki/Desarrollo_de_software) basada en un proceso [iterativo e incremental](http://es.wikipedia.org/wiki/Desarrollo_iterativo_y_creciente) utilizado comúnmente en entornos basados en el [desarrollo ágil de software](http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software) [Schwaber, 2013].

En SCRUM hay tres aspectos fundamentales a diferenciar, los actores, los eventos o acciones y los artefactos. Los actores son los que ejecutarán las acciones para alcanzar los objetivos del proyecto y con ello se genera los artefactos que servirán para planificar o organizar el desarrollo del proyecto.

**Los actores**

**Product Owner:** representa a los *stakeholders* y conoce los requerimientos de alto nivel del producto marcando las prioridades del mismo.

**Scrum Master:** es la persona que asegura el seguimiento de la metodología guiando las reuniones y ayudando al equipo ante cualquier problema que pueda aparecer. Su responsabilidad es entre otras, la de hacer de paraguas ante las presiones externas.

**Scrum Team:** son las personas responsables de implementar la funcionalidad o funcionalidades elegidas por el Product Owner.

**Usuarios o Cliente:** son los beneficiarios finales del producto, y son quienes viendo los progresos, pueden aportar ideas, sugerencias o necesidades [Schwaber, 2013].

**Los eventos o acciones:**

Daily Scrum Meeting: Es una tarea iterativa que se realiza todos los días que dure el Sprint Backlog con el equipo de desarrollo o de trabajo. Es una reunión operativa, informal y ágil, de un máximo de 30 minutos, en la que se hace 3 preguntas a cada integrante del equipo:

* Que tareas se ha realizado desde la última reunión.
* Sobre qué se va a trabajar en el día actual.
* Identificación de obstáculos o riesgos que impiden o pueden impedir el normal avance del proyecto. El Scrum Master debe eliminar cualquier obstáculo que encuentre.

**Sprint Planning Meeting:** es una reunión que tiene por objetivo planificar el Sprint a partir del Product Backlog. El objetivo de esta reunión es la de mover las tareas del Product Backlog al Sprint Backlog. En esta reunión, suelen participar el Product Owner que es quien prioriza las tareas.

Sprint Goal: es un pequeño documento o una breve descripción que indica lo que el Sprint Backlog intentará alcanzar.

***Sprint Review o Demo*:** se revisa en unas 2 horas como máximo el Sprint finalizado. Al llegar a este punto, se debe tener "algo" que el Cliente o el Usuario pueda ver y tocar. En esta reunión, suelen asistir el Product Owner, el Scrum Master, el Scrum Team y personas que podrían estar involucradas en el proyecto. El Scrum Team es quién muestra los avances realizados en el Sprint.

**Sprint Retrospective**: Al finalizar un Sprint Backlog y el Sprint Review, se inicia el **Sprint Retrospective.** El Product Owner revisará con el equipo los objetivos marcados inicialmente en el Sprint Backlog concluido, se aplicarán los cambios y ajustes si son necesarios, y se marcarán los aspectos positivos (para repetirlos) y los aspectos negativos (para evitar que se repitan) del Sprint. [Schwaber, 2013].

**Los artefactos**:

Product Backlog: Corresponde con todas las tareas, funcionalidades y requerimientos a realizar por el equipo de desarrollo. El Product Owner es la persona que se encarga de marcar las prioridades y de mantener y actualizar la lista de tareas.

Sprint Backlog: Corresponde con una o más tareas que provienen del Product Backlog. El Sprint Backlog tiene una duración de 2 a 4 semanas que al momento de iniciar, este no puede ser alterado o modificado en su transcurso. Se tiene que esperar a que concluya el Spring Backlog para realizar la correspondiente modificación o alteración cuyas tareas formaría para del siguiente Spring Backlog. Cuando se ha finalizado un Sprint Backlog, se debe tener un entregable o algo que se pueda mostrar y que enseñe los avances acometidos en el Sprint.

Incremento:

El Incremento es la suma de todos los elementos del Product Backlog completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final de un Sprint, el nuevo Incremento debe estar “Terminado”, lo cual significa que está en condiciones de ser utilizado por el dueño del producto. [Schwaber, 2013].

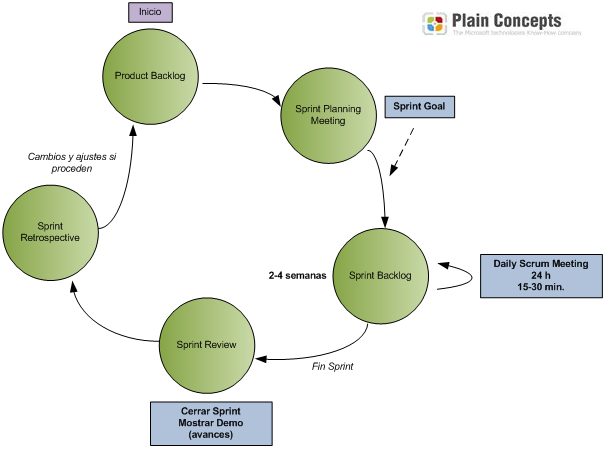
****

Figura 4.1: Flujo de Acciones en un proyecto de Scrum [Schwaber, 2013].

**Planning Poker**

 Es una técnica para calcular una estimación basada en el consenso, en su mayoría utilizada para estimar el esfuerzo o el tamaño relativo de las tareas de [desarrollo de software](http://es.wikipedia.org/wiki/Desarrollo_de_software).

El Planning Poker se basa en una lista de tareas a ser estimado y una baraja de cartas. Un mazo típico contiene tarjetas mostrando la secuencia de Fibonacci incluyendo un cero: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89. Otros mazos utilizan progresiones similares. La razón de utilizar la secuencia de Fibonacci es reflejar la incertidumbre inherente en la estimación. Un mazo que se encuentra en el mercado utiliza la siguiente secuencia: 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, y, opcionalmente, una tarjeta, una con signo de interrogación (?) y otra con signo de infinito (∞), que pueden ser usadas para declarar completa incertidumbre o desconocimiento de la característica. [Piedrasanta, 2011]

Los recursos para trabajar con la técnica Planning Poker son:

* Cartas Planning Poker
* Dos a cuatro participantes
* Un moderador o líder del proyecto.
* Un reloj o algún elemento que determine el tiempo.

**Procedimiento de uso**

En la reunión de la estimación a cada participante se le da un conjunto completo de tarjetas.

La reunión prosigue de la siguiente manera:

* Un moderador, que no jugará, preside la reunión. Puede ser el perfil del Líder de Equipo o Líder de Proyecto.
* El desarrollador con más conocimiento de una determinada característica proporciona una breve introducción sobre la misma. El equipo tiene la oportunidad de hacer preguntas y discutir para aclarar los supuestos y riesgos. Un resumen de la discusión es registrado por el moderador.
* Cada persona coloca una tarjeta boca abajo que representa su estimación. Las unidades utilizadas pueden ser variadas y definidas previamente. Pueden ser días de duración, días ideales o puntos de la historia. Durante el debate, los números no debe ser mencionados en absoluto.
* Todo el mundo muestra sus tarjetas de forma simultánea.
* A las personas con estimaciones altas y bajas se les da un tiempo para ofrecer su justificación para la estimación y la discusión continúa. Como máximo 15 minutos.
* Se repita el proceso de cálculo hasta que se alcance un consenso. El programador que probablemente tenga el entregable tiene una gran parte del voto de consenso, aunque el moderador puede negociar el consenso.
* Se puede utilizar un reloj de arena para asegurar que el debate sea estructurado. Una ronda de Poker finaliza cuando se llego a un acuerdo sobre la estimación o se termino el tiempo del reloj. Se puede dejar una funcionalidad sin consenso y volver después de terminar con las demás tareas. [Piedrasanta, 2011]

**Aplicación de la Metodología de Desarrollo**

**Actores:**

De definió los roles de Scrum de la siguiente manera

|  |  |
| --- | --- |
| **SCRUM** | |
| **Product Owner** | Pablo Chávez(Dueño del Restaurante). |
| **Scrum Master** | Elizabeth Reinoso. |
| **Scrum Team** | Martin Carrizo y Walter Vale. |
| **Usuarios o Cliente** | Personas con discapacidad auditiva. |

**Product Backlog:**

Las tareas del Product Backlog se definieron en una reunión con el Product Owner, Scrum Master y Scrum Team. El Product Owner es quién dio a conocer los requerimientos de alto nivel y priorizó las tareas a desarrollarse. Todas las dudas del Scrum Team acerca de la ambigüedad de los requerimientos fueron disipadas por Product Owner. La reunión fue precedida y coordinada por el Scrum Master. La Tabla 4.1 fue el resultado de la reunión.

La tabla 4.1 está compuesta por la columna ID para identificar unívocamente las tareas del Product Backlog y la del Spring Backlog. El ESFUERZO ESTIMADO para finalizar una tarea fue calculado utilizando Planning Poker. El CRITERIO DE ACEPTACIÓN es la condición que debe lograrse para que la tarea adopte la categoría de finalizada o terminada.

|  |  |  |  |
| --- | --- | --- | --- |
| **PRODUCT BACKLOG ITEMS** | | | |
| **ID** | **TAREA** | **ESFUERZO ESTIMADO** | **CRITERIO DE ACEPTACIÓN** |
| 0 | Anteproyecto | 72 | Se debe formular un documento según los estándares definidos por el establecimiento. |
| 1 | Investigación y elección de las herramientas para el desarrollo del prototipo | 20 | Investigar las herramientas, Framework y librerías de realidad aumentada más óptimas para el desarrollo del prototipo. Realizar una comparación de las tecnologías y justificar la elección en un documento. |
| 2 | Instalación de las Herramientas para el desarrollo del prototipo. | 5 | Realizar la Instalación de las herramientas necesarias y documentar los pasos de instalación. |
| 3 | Uso y manejo de la interfaz de Blender. | 21 | Se debe alcanzar los conocimientos en el manejo de la interfaz de la herramienta de modelado 3D con el objetivo de modificar los modelos ya existentes en internet. |
| 4 | Uso y manejo de la interfaz de Unity 3D. | 20 | Se debe lograr los conocimientos mínimos en el manejo de la interfaz con el fin de crear una aplicación para Android. |
| 5 | Aprendizaje del SDK de Vuforia | 16 | Se debe estudiar la arquitectura del SDK de Vuforia y la programación básica en el uso de sus funciones. También la creación de una base de datos de imágenes para el reconocimiento de marcadores. |
| 6 | Introducción al lenguaje de programación C# y la plataforma .net de Visual Studio 2010. | 12 | Se debe alcanzar los conocimientos básicos sobre el lenguaje de programación C# y del manejo de interfaz de la IDE de Visual Studio 2010. Manejo en la conexión con la base de datos Mysql, NHibernate para la persistencia de datos, uso de formularios y patrones de diseño como Dao y Facade. |
| 7 | Investigación de la simbología de lenguaje de señas. | 8 | Se debe investigar de fuentes bibliográficas la simbología del lenguaje de señas y de profesionales especializados en el tema. |
| 8 | Selección del lugar para el despliegue del prototipo. | 2 | Se debe contar con un lugar físico para la instalación del sistema y posteriormente realizar las prueba de aceptación con personas con capacidades diferentes. |
| 9 | Obtención y especificación de requerimiento funcional y no funcional. | 20 | Obtener una descripción completa del comportamiento del sistema a través del Product Owner en lenguaje natural y realizar una especificación de los requerimientos obteniendo así los siguientes diagramas: Diagrama de caso de uso, Diagrama de contexto, Diagrama de base de datos y diagrama de clases. |
| 10 | Croquis de la interfaz del sistema y de la carta gourmet. | 6 | Diseñar todas las pantallas del sistema para el modulo cliente (Sistema móvil) y para el modulo de administración del sistema (Servidor del sistema). |
| 11 | Documentación | 18 | Se debe documentar lo realizado y volcarlo en sus respectivos capítulos. |
| 12 | Diseño de la arquitectura de software. | 7 | Se debe crear un diseño de alto nivel de la estructura, funcionamiento e interacción entre las partes del software. |
| 13 | Diseño gráfico de la carta de gourmet. | 14 | Se debe realizar el diseño de la carta gourmet en formato digital. Debe ser atractiva visualmente y contar con la combinación equilibrada de textos e imágenes. El diseño debe ser aprobado por el Product Owner. |
| 14 | Diseño de las interfaces del sistema. | 7 | Se debe diseñar el icono, botones y splash de la aplicación para celular. |
| 15 | Elección de la base de datos y conexión con teléfonos móviles. | 10 | Se debe investigar la compatibilidad de las base de datos para el prototipo. Se debe realizar la conexión entre la base de datos y herramienta Unity 3D como primer paso. Como segundo paso realizar la conexión del teléfono móvil con la base de datos local. |
| 16 | Creación de base de datos para el sistema | 3 | Se debe crear la base de datos con todos los privilegios correspondientes en función del diagrama de base de datos. |
| 17 | Configuración del servidor local. | 13 | Se debe configurar el servidor y dejarlo listo para recibir peticiones mediante un cliente. El cliente debe ser un teléfono móvil que envía peticiones para almacenar y recibir datos. Estos datos deben estar alojados en una base de datos en el servidor. |
| 18 | Alta de Pedido | 13 | El cliente puede realizar un pedido utilizando un teléfono celular. El pedido tiene que ser almacenado en la base de datos del servidor. |
| 19 | Modificación de Pedido | 7 | El modulo cliente debe ofrecer la posibilidad de modificar el pedido a través de las pantallas del dispositivo móvil. |
| 20 | Consulta de Pedido por Cliente | 8 | El cliente podrá consultar el pedido realizado a través del teléfono celular. |
| 21 | Consulta de Diarios Digitales | 11 | El cliente tendrá acceso a los diarios digitales mas leídos de Jujuy. |
| 22 | Reproducir video de recomendación de chef | 15 | El cliente podrá reproducir un video de las recomendaciones del chef o bien de las promociones del restaurante. |
| 23 | Navegación por la carta gourmet | 15 | El cliente podrá navegar por la carta gourmet apreciando los diferentes platos o modelo 3D. Debe incorporarse botones para la navegación por categoría y para el desplazamiento por los ítems de la carta gourmet. |
| 24 | Login Administrador | 8 | El administrador o dueño del restaurante podrá ingresar al sistema usando un usuario y contraseña. |
| 25 | Documentación | 15 | Se debe documentar lo realizado y volcarlo en sus respectivos capítulos. |
| 26 | Informe de Avance de Proyecto | 12 | Elaborar el informe de avance de proyecto final. |
| 27 | Consulta de Pedido de Administrador | 13 | El administrador del sistema podrá realizar una consulta de los pedidos realizados por sus clientes. La consulta se realizará entre un intervalo de fechas dado por el administrador. |
| 28 | Imprimir consulta de Pedido | 20 | El administrador del sistema podrá generar un reporte de la consulta de un pedido realizado. Este reporte podrá ser impreso para ser utilizado por el chef en la elaboración de los platos. |
| 29 | Agregar video | 7 | El administrador del sistema podrá agregar un nuevo video sobre las recomendaciones del chef. El alta de los videos debe almacenarse en la base de datos del servidor. |
| 30 | Eliminar video | 4 | El administrador del sistema podrá eliminar un video de la base de datos si lo requiere. |
| 31 | Modificar el precio del plato | 6 | El administrador del sistema podrá modificar el precio del plato de la carta gourmet. Únicamente el campo precio del formulario debe ser modificado. |
| 32 | Buscar una solución al problema de la orden por numero de mesa. | 5 | Al realizar un pedido, el sistema debe identificar unívocamente de que mesa proviene la orden. Se debe implementar una solución utilizando técnica de realidad aumentada. |
| 33 | Agregar simbología de lenguaje de señas al prototipo | 13 | Agregar esta simbología a la aplicación mediante imágenes, videos, etc. |
| 34 | Corrección de errores del sistema. | 15 | Se debe depurar los errores del sistema tanto para el modulo cliente como para el modulo de Administración. Se debe realizar las pruebas de integración y prueba de aceptación en el entorno final. Al finalizar esta tarea, el sistema debe estar listo para la entrega al Product Owner. |
| 35 | Manual de usuario. | 10 | Se dejará listo un manual de usuario para modulo cliente y el modulo de administración al finalizar esta tarea. |
| 36 | Generación de los entregables del sistema. | 4 | Se debe generar los ejecutables para el modulo cliente “apk” y el módulo de administración “exe”. Debe estar incluido en un CD con toda la documentación correspondiente. |
| 37 | Refinado de la documentación | 20 | Se debe refinar la documentación del proyecto final. |
| 38 | Preparación de la defensa fina | 12 | Las diapositivas para la defensa final debe estar terminar al finalizar esta tarea. |
| 37 | Total | 507 |  |

Tabla 4.1: Listado de tareas del Product Backlog

**Sprint Backlog**

Cada iteración del Sprint Backlog fue definido en una reunión con el Product Owner, el Scrum Team y Scrum Máster. Al finalizar la iteración, se realizó una demo mostrando los resultados al Product Owner. Esta reunión es llamada Sprint Review.Se definió una venta de tiempo de aproximadamente 1 mes para finalizar cada Sprint dicando 8 horas /semanales por alumno.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | **Id** | **Tarea** | **Esfuerzo Estimado** | **Asignación** | **Resultado** |
| **Sprint 01**  **13/11/13 al 12/12/13** | 1 | Investigación y elección de las herramientas para el desarrollo del prototipo | 20 | Walter | Finalizado |
| 2 | Instalación de las Herramientas para el desarrollo del prototipo. | 5 | Martin | Finalizado |
| 3 | Uso y manejo de la interfaz de Blender. | 8 | Ambos | Finalizado |
| 4 | Uso y manejo de la interfaz de Unity 3D. | 12 | Ambos | Finalizado |
| 5 | Aprendizaje del SDK de Vuforia | 16 | Martin | Finalizado |
| TOTAL | | 61 | | |
| **Sprint 02**  **07/02/14 al 18/03/14** | 6 | Introducción al lenguaje de programación C# y la plataforma .net de Visual Studio 2010. | 12 | Walter | Finalizado |
| 7 | Investigación de la simbología de lenguaje de señas. | 8 | Martin | Finalizado |
| 8 | Selección del lugar para el despliegue del prototipo. | 2 | Ambos | Finalizado |
| 9 | Obtención y especificación de requerimiento funcional y no funcional. | 20 | Ambos | Finalizado |
| 10 | Croquis de la interfaz del sistema y de la carta gourmet. | 6 | Martin | Finalizado |
| 11 | Documentación | 18 | Ambos | Finalizado |
| TOTAL | | 66 | | |
| **Sprint 03**  **31/04/14 al 30/05/14** | 12 | Diseño de la arquitectura de software. | 7 | Martin | Finalizado |
| 13 | Diseño gráfico de la carta de gourmet. | 14 | Walter | Finalizado |
| 14 | Diseño de las interfaces del sistema. | 7 | Walter | Finalizado |
| 15 | Elección de la base de datos y conexión con teléfonos móviles. | 10 | Walter | Finalizado |
| 16 | Creación de base de datos para el sistema | 3 | Walter | Finalizado |
| 17 | Configuración del servidor local. | 13 | Martin | Finalizado |
| 18 | Alta de Pedido | 13 | Martin | Finalizado |
| TOTAL | | 67 | | |
| **Sprint 04**  **9/06/14 al 1/07/14** | 19 | Modificación de Pedido | 7 | Ambos | Finalizado |
| 20 | Consulta de Pedido por Cliente | 8 | Walter | Finalizado |
| 21 | Consulta de Diarios Digitales | 11 | Ambos | Finalizado |
| 22 | Reproducir video de recomendación de chef | 15 | Martin | Finalizado |
| 23 | Navegación por la carta gourmet | 15 | Walter | Finalizado |
| 24 | Login Administrador | 8 | Martin | Finalizado |
| TOTAL | | 64 | | |
| **Sprint 05**  **2/07/14 al 08/08/14** | 25 | Documentación | 15 | Ambos | Finalizado |
| 26 | Informe de Avance de Proyecto | 12 | Ambos |  |
| 27 | Consulta de Pedido de Administrador | 13 | Martin | Finalizado |
| 28 | Imprimir consulta de Pedido | 20 | Walter | Finalizado |
| 29 | Agregar video | 7 | Martin | Finalizado |
| TOTAL | | 67 | | |
| **Sprint 06**  **11/08/14 al 09/09/14** | 30 | Eliminar video | 6 | Walter | Finalizado |
| 31 | Modificar el precio del plato | 6 | Martin | Finalizado |
| 32 | Buscar una solución al problema de la orden por número de mesa. | 5 | Walter | Finalizado |
| 33 | Agregar simbología de lenguaje de señas al prototipo | 13 | Martin | Finalizado |
| 34 | Corrección de errores del sistema. | 16 | Ambos | Finalizado |
| 35 | Manual de usuario. | 10 | Walter | Finalizado |
| 36 | Generación de los entregables del sistema. | 4 | Martin | Finalizado |
| TOTAL | | 60 | | |
| **Sprint 07**  **10/09/14 al 11/10/14** | 37 | Refinado de la documentación | 20 | Ambos | Finalizado |
| 38 | Preparación de la defensa fina | 12 | Ambos | Finalizado |
| TOTAL | | 32 | | |

Sprint Retrospective: En la tabla 4.3 se detalla las Retrospective de los 7 Sprint realizados.

|  |  |  |  |
| --- | --- | --- | --- |
| Retrospective | | | |
| **SPRINT** | **A favor** | **En contra** | **A mejorar** |
| **Sprint 01** | **\***Se obtuvo buenos resultados en las investigación realizadas, pudiendo definir con certeza las tecnologías a utilizar.  \*Contar con el equipamiento necesario como PCs y celulares con sistema operativo Androide. | \*Falta de comunicación y organización en cuanto a la ejecución de las tareas.  \*No contar con un lugar físico con los servicios adecuados como internet. | \*Implementar las daily para mejorar la comunicación y organización de las tareas.  \*Buscar en casas de amigos y familiares un lugar adecuado para el desarrollo del proyecto. |
| **Sprint 02** | \* Conocimientos en los lenguajes de programación orientado a objetos como java o C#.  \* Los miembros del Scrum Team ya cuenta con la experiencia en el uso de Scrum.  \*Comenzar en etapas temprana la redacción del marco teórico del proyecto final. | \* Se dedico demasiado tiempo en completar las tareas asignadas (superior a las estimadas). | \*Definir con mayor detalles las tareas a realizar por el equipo. |
| **Sprint 03** | \*Contar con experiencia en el manejo de herramientas de diseño como Adobe Photoshop para la elaboración de interfaces gráficas.  \*Mejora de la Comunicación y trabajo en equipo. | \*No estar de acuerdo en el diseño de las interfaces del sistema.  \*No contar con un sistema de control de versiones y de almacenamiento de archivos. | \*Definir la interfaz a priori para no llegar a la desavenencia en cuanto al diseño.  \*Se debe implementar SVN para la gestión y versión de archivos para el proyecto |
| **Sprint 04** | \*Uso de patronos de diseño para mejorar la arquitectura del software. | \*Falta de documentación del código. | \*Documentar el código al terminar un método o clase. |
| **Sprint 05** | \*Cumplir con las tareas asignadas en tiempo y forma según la planificación. | \*Las demoras administrativas en cuanto a la aprobación del anteproyecto. | \*Buscar información del estado del anteproyecto en Secretaría Académica de forma periódica. |
| **Sprint 06** | \* Disponer de personas con discapacidad auditiva para realizar las pruebas de aceptación.  \* Revisión de Código para la optimización de recursos del sistema. | **Llenar** | **Llenar** |
| **Sprint 07** | **Llenar** | **Llenar** | **Llenar** |
|  |  |  |  |
|  |  |  |  |

Tabla 4.2: