# CAPITULO VII – SEGUNDA ITERACÍON

## La metodologia scrum es iterativa e incremental, por lo tanto este capitulo, abarca todo lo relacionado al desarrollo de la segunda iteracion.

## 6.1 Planificación de la iteración

Para la planificación de la segunda iteración, se ha tomado en cuenta los puntos pendientes o la historia de usuario no aceptada en la primera iteración. Dicha historia de usuario fue introducida en la pila del producto de la segunda iteración.

### 6.1.1 Pila de la Iteración

Para el desarrollo de la segunda iteración del proyecto, se ha tomado la decisión de desarrollar las siguientes historias de usuario, tomando en cuenta la prioridad y complejidad. Dichas historias de usuario se pueden ver en la tabla 7.1.

|  |  |  |
| --- | --- | --- |
| **Numero Historia de Usuario** | **Nombre** | **Prioridad** |
| 2 | Registro de reserva regular | Alta |
| 4 | Yo como administrador me gustaria registrar reserva administrativa. | Alta |
| 6 | Como administrador me gustaria realizar reservas periódicas con las Reservas Especiales y administrativas | Alta |

### 6.1.2 Estimación de esfuerzo

La estimación de las historias de usuario se las realizo de cuerdo a la serie Fibonacci, tomando como 1 la menor complejidad y 8 mayor complejidad.

|  |  |  |
| --- | --- | --- |
| **Numero Historia de Usuario** | **Nombre de Historia de Usuario** | **Estimación** |
| 2 | Registro de reserva regular | 8 puntos |
| 4 | Yo como administrador me gustaria registrar reserva administrativa. | 5 puntos |
| 6 | Como administrador me gustaria realizar reservas periódicas con las Reservas Especiales y administrativas | 8 puntos |

### 6.1.3 Definición de tareas

A continuación, se definen las tareas necesarias para cumplir como las historias de usuario planificadas.

|  |  |
| --- | --- |
| **Numero Historia de Usuario** | **Tareas** |
| 2 | * Diseñar la base de datos para realizar las reservas. * Agregar tablas a la base de datos para registro de reservas. * Implementar los servicios en Angular Js. * Implementar los servicios en CodeIgniter para el registro de reservas. * Implementar la integración de los servicios para la funcionalidad. * Realizar las pruebas correspondientes. |
| 4 | * Modificar la base de datos para poder realizar las reservas administrativas. * Diseñar e implementar el formulario para el registro. * Implementar los servicios en Angular y Codeigniter. * Hacer la integración de la comunicación de cliente y servidor. * Realizar las pruebas. |
| 6 | * Modificar la implementación en los controladores de Angular en reservas regulares. * Modificar los servicios RESTFULL. * Modificar la interfaz para permitir el registro de reservas periódicas. * Implementar la integración entre cliente y servidor. * Realizar pruebas. |

## 6.3 Diseño

El diseño inicial de la base de datos ha sido modificado para la segunda iteración, se han agregado tablas para dar soporte a los registros de reservas regulares, administrativas y periódicas. Dichos cambios, se pueden observar en la figura 7.1.

C:\Users\Beimar\Desktop\segundo sprint.emf

*Figura 6.1: Diseño de la base de datos (Elaboracion propia, 2016)*

## 6.5 Pruebas

Para realizar el control de calidad del desarrollo, se ha utilizado las platillas que se muestran en la parte inferior, donde se observa a detalle los pasos que se deben seguir para los casos de prueba.

Han sido agregados los casos de prueba necesarios para cubrir los criterios de aceptación.

|  |  |
| --- | --- |
| Titulo | Verificar la notificación de registro de campos deportivos. |
| Descripción | Verificar que al realizar un registro de campo deportivo con los datos correspondientes, el sistema debe notificar que el campo deportivo se ha registrado correctamente. |
| Requerimientos | * Navegador web |
| Pasos | 1. Abrir el navegador. 2. Introducir la siguiente url: <http://localhost/compleapp/complejoapp/#/admin/registrarCampo>. 3. Verificar que el formulario de registro se encuentre cargado 4. Llenar el formulario de registro con los datos del campo deportivo. 5. Presionar el botón “Guardar”. 6. Verificar que la notificación de éxito se muestre. |
| Resultado Esperado | La notificación de éxito al registrar un campo deportivo debe mostrarse. |

|  |  |
| --- | --- |
| Titulo | No debería habilitarse el botón de registro, si los campos del formulario no están introducidos. |
| Descripción | El botón de registro de campos deportivos no debería habilitarse, si alguno de los campos del formulario de registro no han sido introducidos previamente. |
| Requerimientos | * Navegador web |
| Pasos | 1. Abrir el navegador. 2. Introducir la siguiente url: <http://localhost/compleapp/complejoapp/#/admin/registrarCampo>. 3. Verificar que el formulario de registro se encuentre cargado 4. No llenar el formulario de registro con los datos del campo deportivo. 5. Intentar presionar el botón “Guardar”. 6. Verificar que el botón “Guardar” no se encuentre habilitado. |
| Resultado Esperado | No se habilita el botón de “Guardar”. |

## 6.6 Revisión de la Iteración

A continuación, se muestra la tabla, en donde se puede observar las historias de usuario que se consideran como terminadas de la segunda iteración y además, resultan como el incremento de la segunda iteración.

Una observación importante, es que en esta segunda iteración ya se tiene un dominio sobre las tecnologías, que dan como resultado un mejor desempeño en esta iteración.

|  |  |  |  |
| --- | --- | --- | --- |
| **Numero H.U.** | **Criterios de Aceptación** | **Cumple** | **Historia Aceptada** |
| 2 | * Debo ser capaz de agregar una reserva con los siguientes datos: nombre cliente, teléfono, campo deportivo, hora, fecha, cantidad de horas. | SI | SI |
| * Debo ser capaz de ver todas las reservas ya realizadas. | SI |
| * No debe registrar una reserva con fecha menor a la actual. | SI |
| * No debe registrar una reserva con fecha actual y hora menor a la hora actual. | SI |
| * No debe registrar una reserva, si existe otra reserva para el mismo campo, el mismo día y exista colisión en horarios. | SI |
| * No debe registrar una reserva, que tenga horario inicio mayor o igual al horario final, | SI |
| * No debe registrar una reserva, cuya duración es menor a una hora. | SI |
| 4 | * Debo ser capaz de resgistrar una reserva administrativa. | SI | SI |
| * Debo ser capaz de ver todas las reservas ya registradas. | SI |
| * No debe registrar una reserva con fecha menor a la actual. | SI |
| * No debe registrar una reserva con fecha actual y hora menor a la hora actual. | SI |
| * No debe registrar una reserva, si existe otra reserva para el mismo campo, el mismo día y exista colisión en horarios. | SI |
| * No debe registrar una reserva, que tenga horario inicio mayor o igual al horario final. | SI |
| * No debe registrar una reserva, cuya duración es menor a una hora. | SI |
| 6 | * Realiza la reserva si está o sus repeticiones no tienen choques con otras reservas y sus repeticiones respectivas y si sus datos son válidos. | SI | SI |
| * Como maximo, se deben realizar solo 10 reservas por un cliente. | SI |
| * Se debe poder especificar de que tipo de reserva periodica se esta registrando. | SI |

### 6.6.1 Grafico BurnDown

Analizando el grafico burndown de la segunda iteración, se puede concluir que se han completado 21 puntos, lo cual indica que las historias de usuario planificadas para la iteración 2, han sido aceptadas satisfactoriamente. Haciendo una comparativa con la primera iteración, se puede observar que se ha aumentado la velocidad de desarrollo.