Universidad Simón Bolívar, sede Sartenejas

Profesores: Alejandro Teruel, José Alfonso Reinoza Contreras

Scrum Master: Prof. Jean Carlos Guzmán (SM)

Materia: Ingeniería de Software (CI-3715)

Repositorio: github.com/ucode3715/IngSoftware

Informe final

<UCode>

Karla Alzuro <08-10037>

Gabriel Álvarez <09-10029>

Carlos Farinha <09-10270>

Francisco Martínez <09-10502>

F. Miguel Saraiva <09-10794>

Ricardo Vethencourt <09-10894>

Caracas, 26 de noviembre de 2014.

**Introducción**

El producto SAGE (Sistema Automatizado de Gestión de Estacionamientos) es lo que resulta por parte del equipo profesorado tras atender una necesidad del mercado que ha sido poco explorada: la de hacer reservas temporales cortas de puestos de estacionamientos. (Existen reservas de estacionamientos, pero se reserva un puesto por mes o año). Es la responsabilidad de éste equipo de desarrollo llevar a cabo las implementaciones necesarias para el debido funcionamiento de un demo de dicho producto, junto a su correspondiente documentación que asegure que el demo puede ser expandido o modificado por otros grupos, para así eventualmente dar con un producto terminado.

El informe que se presenta a continuación cumple con mostrar la documentación previamente mencionada. En el mismo se incluye una descripción detallada del problema, la propuesta que se tiene como solución, se mencionan a las diferentes personas involucradas y sus roles. También incluye lo relacionado a diseño y arquitectura, sus respectivos diagramas, y todo lo relacionado a pruebas y verificaciones.

Es importante notar que como el producto se desarrolla bajo un contexto académico, las herramientas utilizadas no son manejadas anteriormente. Esto con la finalidad de aprender a usarlas, pero también lleva consigo un retraso y entorpecimiento del desarrollo del producto.

**Requisitos**

* **Descripción del dominio:**

Hay dos tipos de clientes a considerar: los usuarios que reservan puestos, y los dueños que registran su estacionamiento en el sistema. Estos segundos son los clientes directos del equipo de desarrollo, son a los cuáles se les debe vender el producto, sin embargo, como hay que implementar partes donde el usuario (es decir, los clientes de nuestros clientes) interactúa con el sistema, se deben considerar a los mismos como directos. Otro elemento importante son los estacionamientos, ellos deben estar representados en el sistema de una manera concisa y robusta, puesto a que son el eje central del funcionamiento físico del servicio que la página ofrece.

Fuera de estos elementos físicos, hay otros dos elementos importantes que se deben manejar con cuidado a la hora de implementar funcionalidades: el tiempo y el dinero. Es importante haber hecho las consideraciones necesarias y establecer claro las reglas de juego a la hora de solicitar una reserva en cierto horario, y también es importante encontrar una representación adecuada para el dinero en el sistema, por razones obvias.

* **Enunciado del problema y propuesta:**

Se desea crear un sistema de reserva multiestacionamiento que permita a usuarios elegir un estacionamiento para reservar un puesto a cierta fecha y en cierto intervalo de tiempo, realizando su pago mediante tarjeta de crédito, con el incentivo de que si ingresan al estacionamiento bajo una reserva, la tarifa a cancelar es menos costosa que la tarifa de clientes sin reserva (para efectos del demo no hay clientes improvistos).

* **Stakeholders:**

Producto:

-Usuarios directos: dueños de estacionamientos y usuarios de los mismos.

-Clientes: dueños de estacionamientos.

-Desarrolladores: equipo UCode.

Proceso:

-Desarrolladores: miembros del equipo UCode.

-Empleador: prof. Teruel, prof. Reinoza (gerentes de desarrollo), prof. Guzmán (Scrum Master)

Negocio:

-La empresa: el curso de ingeniería de sistemas, o en particular el equipo UCode junto al equipo profesorado.

-El cliente: dueños de estacionamientos.

-Como este producto queda dentro del alcance de la materia únicamente, no se presentan profesionales externos ni sociedad como afectados.

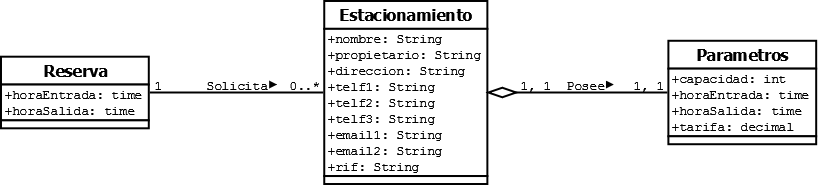
* **Aspectos legales y éticos:**

Como el software se desarrolla en y para el país de Venezuela, se deben seguir los lineamientos de la Ley Especial contra Delitos Informáticos emitida por el Tribunal Supremo de Justicia. Es importante saber manejar de manera segura los contenidos en las bases de datos. Los teléfonos, dirección y correos de los estacionamientos son elementos de baja seguridad; ellos pueden encontrarse de manera pública puesto a que son instrumentos casi publicitarios y no forman parte de información privada. Cualquier otro dato del dueño de estacionamiento, de existir, se debe encriptar en la base de datos para evitar en lo posible el acceso a ellos por parte de personas externas al sistema. De igual manera, los datos de pago de los clientes, así como las mismas reservas, no deben ser de fácil acceso para cualquier usuario que no sea administrador.

Al tratarse de un software desarrollado por ingenieros (en formación), el desarrollo del mismo debe estar alineado con el Código de Etica Profesional del Colegio de Ingenieros de Venezuela, esto incluye la presentación de un producto de calidad que fortalezca la confianza que la sociedad pone en el status de profesional, a un precio justo mas no menor al establecido por el CIV, entre otras cosas.

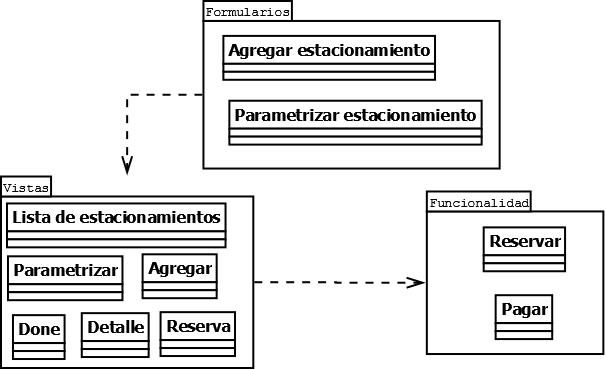
**Diseño**

* **Diagrama de clases:**

****

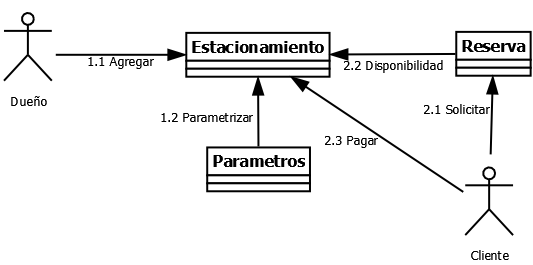
Las clases que conforman al producto, a nivel abstracto, son las de reserva, estacionamiento y parámetros. Cada instancia de parámetros es un agregado a la clase de estacionamiento que, por razones de orden de desarrollo, se agregó después de haber creado los estacionamientos, formando así una funcionalidad y una clase aparte. Cabe mencionar que la relación entre el estacionamiento y su agregado es exactamente de 1 a 1. Si bien pueden haber dos estacionamientos con los mismos parámetros, ellos serían instancias distintas de la clase.

* **Diagrama de paquetes:**

****

Las vistas son aquellos elementos que el usuario visualiza en la página web, y usan los formularios mencionados. Ambas cosas funcionan para llevar a cabo la funcionalidad del servicio que se ofrece.

* **Diagrama de interacción:**

****

Un dueño agrega su estacionamiento al sistema, luego determina sus parámetros. Esto le permite al cliente/usuario solicitar su reserva (en ella indica en cuál estacionamiento va a reservar). Esta clase se encarga de revisar la disponibilidad de un puesto en el horario y estacionamiento solicitado, si hay disponibilidad, el usuario pasa a realizar su pago.

**4. Programación**

|  |  |  |
| --- | --- | --- |
| Sprint | Historias | Realizado |
| 1 | 1. Nuevo estacionamiento. | Completo |
| 2. Diseño de la interfaz de la historia Nuevo Estacionamiento. | Completo |
| 2 | 1. Diseñar e implementar interfaz de Nuevo estacionamiento. | Completo |
|  | 2. Diseño e Implementar interfaz y funcionalidad: Parametrizar estacionamiento. | Completo |
|  | Tareas adicionales:  a.- Mejorar dominio de Django  b.- Manejo de caracteres especiales del Castellano en Django: acentos, diéresis y la “ñ”. | Falto b |
| 3 | 1. Funcionalidad y Diseño de Interfaz : Cambiar entre estacionamientos. Como propietario de varios estacionamientos puede cambiar de vista para ver las características de mis estacionamientos. | Completo |
| 2. Implementar la interfaz de Cambiar entre estacionamientos. | Completo |
| 3. Solicitar reserva (versión inicial). | Completo |
| Tareas adicionales  a.- Manejo de caracteres especiales del Castellano en Django: acentos, diéresis y la “ñ”.  b.- Revisar tipo Decimal en Python. | Falto a |
| 4 | 1. Solicitar reserva (versión inicial). | Completo |
| 2. [INVEST] Pagar solicitud factible de reserva (versión inicial) | Completo |
| Tareas adicionales  a.- Manejo de caracteres especiales del Castellano en Django: acentos, diéresis y la “ñ”. [Pendiente desde el sprint 2]  b.- Revisar tipo Decimal en Python.  c.- Completar la distribución (release) del curso. El release incluye la documentación completa del release, el diagrama de clase del dominio, diagramas de interacción, todas las pruebas unitarias  y de integración del release, el código refactorizado incluyendo el arnés de pruebas. | Falto a |

**5. Verificación**

5.1 Informe de pruebas unitarias: Por desarrollar

5.2 Informe de pruebas de integración: Por desarrollar

**6. Métodos, Técnicas y Herramientas utilizadas:**

          El método principal y fundamental que utilizamos para el desarrollo de nuestro software fue Scrum bajo el mando de un scrumaster que lo realizo un profesor del conjunto docente, podemos mencionar entre las técnicas del desarrollo que utilizamos el Desarrollo Dirigido por Pruebas (TDD) puesto que nos dividimos en pequeñas parejas para la división del trabajo, en este TDD se realizo múltiples veces un protocolo de refactorización para apoyarnos en la perfeccion de un software que posea una vida útil practica, además para ello se elaboraron una serie de pruebas unitarias y de Integración para verificar la fiabilidad y confiabilidad de nuestro software. Para trabajar en ello se pudo utilizar la herramienta de Git como repositorio del equipo, además de PyUnit para los casos de prueba puesto que trabajamos en lenguaje Python dentro de Django.

**Conclusiones y recomendaciones**

La totalidad de las implementaciones del demo que se lograron desarrollar no es suficiente como para presentar este demo a algún cliente potencial. Sin embargo, es necesario considerar que este trabajo se desarrolla bajo un contexto académico, y el propósito principal del mismo es aprender a usar las herramientas mencionadas así como conocer y poner en práctica los conceptos que fueron enseñados durante la teoría. Con este informe, y en conjunto con la presentación del producto final hecho, se demuestra que dichos objetivos principales fueron alcanzados.

De la misma manera que éste equipo lo hizo, se recomienda a cualquiera que quiera retomar el proyecto que utilice la herramienta Bootstrap en conjunto con Django, esto facilita el uso de manera relativamente sencilla de elementos css y html para el desarrollo de la página. Lo que resulta en un producto estéticamente agradable y reduce el esfuerzo necesario.