Propuesta de Solución

**Parqueadero de múltiples niveles por medio de una grúa**

**Autor:**

**Luis Miguel Ponce de leon Lopera**

**Docente de Fundamentos de Programación:**

**Alberto Restrepo Velasquez**

Medellín, Abril 2013

# Introducción

Este documento se presenta como la segunda entrega del trabajo final de la materia Fundamentos de programación. Esta entrega hace parte del espacio del problema del proyecto, y busca aplicar las prácticas que se enseñan en la materia Principios de desarrollo de software (La cual curse el semestre 2012-2 en EAFIT como estudiante de convenio colegio).

En esta entrega se presenta el diseño de la solución a la práctica de fundamentos aplicando los elementos de la programación orientada a objetos enseñados en dicho espacio. Se presenta un diagrama de clases, que describe las entidades y sus relaciones dentro del sistema. También se presentan otros elementos con los cuales en conjunto con el diagrama de clases serán el punto de partida para la implementación.

**Nota**: Al final del documento se anexan todos los diagramas a mayor escala para facilitar su lectura.

# Diseño de la solución

## Estructura del sistema

## 

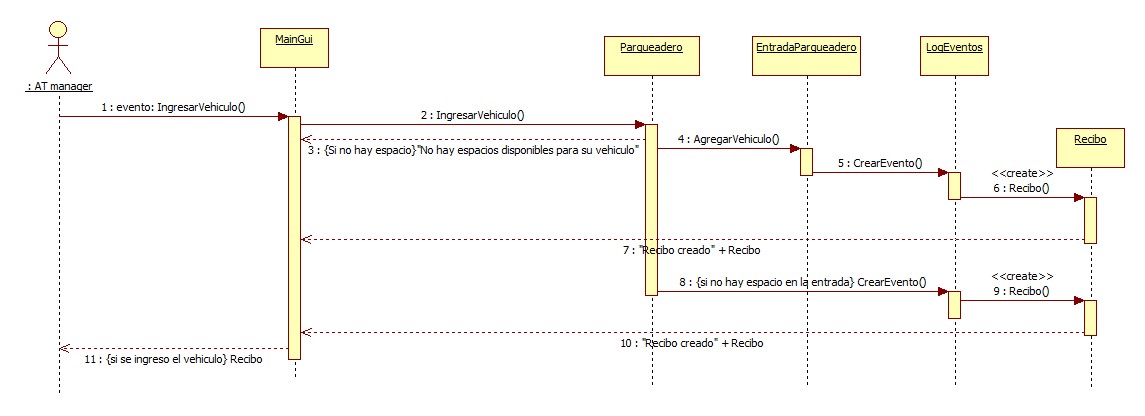
|  |  |
| --- | --- |
| **Nombre de la clase** | **Descripción general de la responsabilidad de la clase dentro del sistema** |
| **Parqueaderos** | Clase encargada de administrar los parqueaderos de la torre y el sótano. Esta clase se encarga de ubicar los vehículos en sus entradas de acuerdo a su placa. Esta es el puente entre la información suministrada por la MainGui y las diferentes entradas del parqueadero. |
| **EntradaParqueadero** | Esta es la encargada de almacenar los vehículos dentro de las celdas que componen la entrada. Esta también es la que al ingresar un vehículo, o bien al retirarlo, debe organizar el orden de la entrada, teniendo en cuenta que el vehículo que se espera salga más rápido este de primero. |
| **Vehiculo** | Es la clase encargada de mantener la información relacionada con el vehículo que es ingresado y almacenado. |
| **TipoDeVehiculo** | Clase que contiene los 3 posibles tipos de vehículos y los precios por hora de estos. |
| **LogEventos** | Clase que sigue el patrón de diseño singleton para que solo exista una instancia de esta. Es la encargada de crear los recibos que identifican los eventos que ocurren en el sistema, de almacenarlos y administrarlos. |
| **Recibo** | Clase que almacena la información relacionada con un evento. Este es el que genera el cobro y si es necesario le genera la deuda acumulada al cliente. |
| **Cliente** | Clase que almacena la información relacionada con un cliente determinado y que guarda la deuda que posee este. |
| **RelojInterno** | Clase encargada del paso constante del tiempo, del control de este y de permitir que se adelante a discreción del usuario. |
| **HoraDelDia** | Clase que permite almacenar una hora con sus minutos siguiendo el formato 24horas. Además debe permitir realizar comparaciones entre estas. |
| **MainGui** | Sirve como el puente entre los elementos dela Gui y las demás clases del sistema que se encargan del manejo como tal del sistema. Se espera que esta inicie los métodos de las diferentes clases cuando su respectivo elemento de la Gui sea activado (como en el caso del ingreso de vehiculo con el botón de ingreso). |

## Asignación de responsabilidades

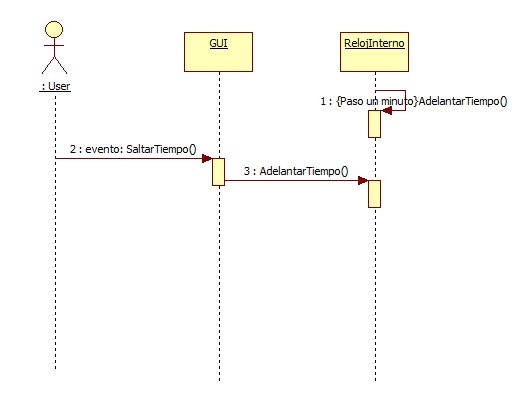
|  |  |  |
| --- | --- | --- |
| **Nombre del evento externo: Ingreso Vehículo al Parqueadero** | | |
| **Acciones a realizar por el sistema** | **Clase responsable de ejecutar la acción** | **Clase con la cual colabora** |
| 1. Recibir el evento y la información del evento, esto incluye: La placa del vehículo, tipo del vehículo, tiempo estimado de parqueo y el ID del cliente. Además se obtiene la hora actual del sistema. | MainGui | Parqueaderos,  RelojInterno |
| 1. Se Obtiene la entrada en que debe ser ingresado el vehículo por medio de la placa. | Parqueadero | Vehiculo |
| 1. Si la entrada asignada tiene espacios disponibles el vehículo ingresa a esta, de no ser así ingresa al sótano. En el caso de que no se pueda ubicar ni en su entrada ni el sótano, el ingreso se detiene y se le informa al cliente que no hay parqueaderos disponibles. | Parqueadero | EntradaParqueadero, MainGui |
| 1. Ubicar el vehículo dentro de su entrada en una celda teniendo en cuenta la hora estimada de retiro. | EntradaParqueadero | Vehiculo |
| 1. Se crea el recibo relacionado con el evento, con la información recibida anteriormente y de ser necesario se crea el usuario en el sistema. | LogEventos | MainGui,  Recibo |

|  |  |  |
| --- | --- | --- |
| **Nombre del evento externo: Incremento del tiempo en el reloj** | | |
| **Acciones a realizar por el sistema** | **Clase responsable de ejecutar la acción** | **Clase con la cual colabora** |
| 1. Se recibe el llamado al evento de incremento, sea porque paso un minuto o porque el usuario pide un incremento del tiempo. | RelojInterno | MainGui |
| 1. Se hace el incremento en horas y minutos del tiempo almacenado como tiempo actual (evitando que se pase de 24h y 60m). | RelojInterno |  |
| 1. Se actualiza el tiempo actual del RelojInterno. | RelojInterno |  |

## Colaboraciones entre instancias de objetos

****Ingreso de vehículo al parqueadero:**

**Incremento del tiempo en el reloj:**

**

# Avance en la implementación

*[Junto con el trabajo escrito, se debe mostrar avance en la implementación del sistema que tenga: Definición en Java de las clases identificadas en el espacio del problema, con los atributos definidos y la declaración de los métodos requeridos para los dos escenarios que se analizaron. Implementación completa del escenario de* **Incremento del Tiempo en el Reloj***.*

*Nota: No se exige que la implementación de esta entrega tenga ya la Interfaz de Usuario; se puede tener una clase general (por ejemplo* ***ManejadorTorre****) que sirva como una clase “puente” que recibe los eventos externos.*

# Conclusiones

En este trabajo se trabajaron elementos necesarios en diseño de la solución de un problema relacionado con el diseño de software. Se realizó un diagrama de clase para identificar las clases que deberán ser creadas en el espacio de la solución, además se identificaron las relaciones entre las diferentes entidades. También se identificaron las acciones necesarias para llevar a cabo dos procesos necesarios de la solución del problema.

El proceso realizado en esta entrega es fundamental para el correcto desarrollo de un proyecto que involucre el desarrollo de software, en especial si piensa trabajar con un equipo de desarrolladores, donde todos deben de tener claras las “reglas de juego” y saber cómo se piensa diseñar el software y su arquitectura.

Al realizar esta entrega no se presentó ninguna dificultad, pues realizar lo que se entrega en este documento debe convertirse en una segunda naturaleza para un buen desarrollador de software y no debe presentar grandes dificultades.