Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Programación Orientada a Objetos

Prof.: Mauricio Avilés

Proyecto I

Parqueo Vehicular

Tomas Apeltauer

Adrián Rodríguez

Saúl Zamora

Semestre I

2014

# Resumen ejecutivo

Todos somos conscientes de que el sistema vial de la Costa Rica actual no fue pensado para que soportara el alto volumen de vehículos que hoy día circulan en nuestras carreteras, y debido a estas fallas de diseño, nos vemos forzados a optar por opciones alternativas a la ampliación de las carreteras. El uso de transporte público y/o alternativo para disminuir el flujo vehicular y la implementación de parqueos. Este proyecto se basa en la segunda alternativa.

# Introducción

Con el pasar de los años, todos hemos notado el aumento en el flujo vehicular en las carreteras de nuestro país. Esto ha ocasionado el aumento en los congestionamientos de tránsito (presas) y en verdaderas batallas en los parqueos y en las orillas de las carreteras cuando los conductores se disponen a parquear sus vehículos.

Dada la urgencia con la que se requiere de una solución y de la enorme dificultad que presenta un rediseño y ajuste de la red vial nacional en su totalidad, sumada a la inseguridad que se asocia con dejar un automóvil en la vía pública, ha convertido en los parqueos vehiculares privados en una alternativa muy popular. Un parqueo vehicular es un local comercial que presta el servicio de estacionamiento para vehículos automotores. La lógica del negocio es muy simple: cuenta con una serie de espacios donde los clientes pueden estacionar su vehículo, garantizando parqueo y seguridad a cambio de una tarifa por hora.

En este proyecto se implementa un sistema informático que ayuda con la administración de parqueos; en el cual el usuario (administrador y/o empleado) pueda llevar cuenta del espacio disponible con el que cuenta, la cantidad de dinero del que dispone en la caja registradora y un historial con los recibos de los automóviles que han estado en el parqueo.

# Análisis del problema

El sistema debe ser capaz de solventar las funcionalidades básicas requeridas para el manejo de un parqueo:

* Autorización y manejo de credenciales, diferenciación de usuarios (normal/administrador)
* Proveer seguridad, manejo de contraseñas y encriptación
* Brindar un sistema confiable para manejar las finanzas del parqueo, cobro a los clientes y manejo de vueltos, monto total al iniciar y finalizar el día
* Mostrar el estado actual del parqueo, espacios ocupados y disponibles
* Historial de facturas, poder ver todos los recibos realizados desde que se abrió el parqueo por primera vez

Por medio de archivos XML se pretende solventar las funcionalidades de historial y administración de cuentas. Las finanzas del parqueo se manejarán por medio de las bibliotecas matemáticas de JAVA

# Análisis de resultados

|  |  |
| --- | --- |
| Estructura | Resultado |
| Abrir parqueo | A |
| Credenciales de Usuario | A |
| * Contraseñas | A |
| Ingresar vehículo | A |
| Salida vehículo | A |
| * Generar factura | D |
| Estado del parqueo | D |
| Historial de facturas | D |
| Caja | D |
| * Entrada de efectivo | D |
| * Salida de efectivo | D |
| Cerrar parqueo | D |
| Configuración de parámetros | A |
| Salida del programa | D |

Simbología

* A = Terminado con éxito
* B = Terminado con problemas pequeños
* C = Terminado con problemas grandes
* D = Implementado solo en diseño
* E = No implementado

# Recomendaciones

* El uso de un lenguaje diferente, como C#, puede hacer más fácil y simple el manejo de archivos XML.
* La implementación de Bases de Datos para el manejo de registros (historial de facturas, credenciales de usuario) resulta más eficiente que el manejo de archivos XML.

# Referencias

* MVC Java Turorial
  + <https://www.youtube.com/watch?v=dTVVa2gfht8>
* Examples of Model-View-Controller Pattern
  + <http://www.cs.utsa.edu/~cs3443/mvc-example.html>
* The Simplest Model View Controller (MVC) Java Example. Mack, Joseph. (2011)
  + <http://www.austintek.com/mvc/>
* MVC Java Tutorial. Banas, Derek. (2013)
  + <http://www.newthinktank.com/2013/02/mvc-java-tutorial/>
* Suggestions for library to hash passwords in Java
  + <http://stackoverflow.com/questions/2860943/suggestions-for-library-to-hash-passwords-in-java/11038230#11038230>
* Which exception to throw if list is empty in Java?
  + <http://stackoverflow.com/questions/11201420/which-exception-to-throw-if-list-is-empty-in-java>
* Destroy Jframe object
  + <http://stackoverflow.com/questions/15733587/destroy-jframe-object>
* How can I set .gitgnore to ignore all .jars but not in the expected folder
  + <http://stackoverflow.com/questions/14021400/how-can-i-set-gitignore-to-ignore-all-jars-but-not-in-the-expected-folder>
* Java – Date & Time. (2014)
  + <http://www.tutorialspoint.com/java/java_date_time.htm>
* How do I test to see if the enter key is pressed in a textfield in Java?
  + <http://stackoverflow.com/questions/10745156/how-do-i-test-to-see-if-the-enter-key-is-pressed-in-a-textfield-in-java>
* Is Java realle passing objects by value? [duplicate]
  + <http://stackoverflow.com/questions/7893492/is-java-really-passing-objects-by-value>