

IC-2101 Programación Orientada a Objetos - Prof. Mauricio Avilés

Tarea Programada I – Parqueo Vehicular

Introducción

Actualmente las ciudades de Costa Rica, en especial San José, muestran un congestionamiento vehicular problemático. El aumento de la cantidad de vehículos circulantes, sumado a una planificación deficiente por parte del Estado, han convertido a las carreteras en verdaderos campos de batalla donde atravesar una distancia de 20 km puede llegar a tomar cerca de 2 horas. Esto ha obligado a las autoridades a tomar algunas medidas para asegurar la circulación de los vehículos, como lo es la reducción de espacios de estacionamiento en la vía pública. Encontrar un espacio disponible para estacionar un automóvil es una tarea difícil de lograr.

Esta situación, 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.

El actual proyecto consiste en la elaboración de una aplicación que permita a un local de este tipo administrar las entradas y salidas de vehículos, así como la facturación del servicio.

Software a desarrollar

Su trabajo consiste en implementar una aplicación de escritorio para administrar un parqueo utilizando el lenguaje Java y la Programación Orientada a Objetos.

Valores iniciales y autorizaciones

Al iniciar la ejecución, la aplicación debe cargar algunos datos generales que serán utilizados durante su funcionamiento: nombre del local, eslogan, teléfono, cédula jurídica, horario de atención, tarifa por hora, número secuencial de la siguiente factura, monto mínimo de efectivo en caja para abrir y contraseña de administrador encriptada. La aplicación lleva control de la cantidad de efectivo que hay en la caja del establecimiento, pero este monto se inicializa siempre con cero. Para incluir o retirar dinero de la caja, se hace un movimiento de efectivo; y para realizarlo debe solicitarse y verificarse la contraseña de administrador.

Los datos generales y la contraseña encriptada deben almacenarse en un archivo XML. El formato del archivo y el método de encriptación de la contraseña a utilizar son decisiones que deben tomarse como parte del trabajo.

El archivo también debe guardar la cantidad de espacios disponibles en el parqueo y la descripción de cada uno. Esta información debe cargarse al inicio en la aplicación y debe permitirse modificarla solamente si el parqueo se encuentra cerrado.

Abrir el parqueo

Una vez inicializados los valores de la aplicación, el parqueo puede abrirse. Para abrir el parqueo debe cumplirse que estén definidos los espacios del parqueo, y que se cumpla con el monto mínimo de efectivo en la caja. Si no se cumple, debe mostrarse un mensaje de error para informar al usuario que debe ingresar a la sección de configuración de parámetros.

Ingreso de vehículos

Cuando un vehículo ingresa al parqueo debe anotarse su número de placa (obligatorio) y demás características del automóvil (color, marca, modelo), y la hora de ingreso del mismo al estacionamiento. Debe registrarse también el espacio en el que el vehículo fue estacionado. Se le entrega al cliente un ticket que contiene esta información. A continuación se muestra un ejemplo del ticket:

Parqueo Morazán

Abierto las 24 horas. Tel: 48657348.

Placa:

Hora de entrada:

Este ticket debe imprimirlo el sistema con los datos ya completos. Para efectos de este proyecto, el ticket no se va a imprimir, si no que en lugar de eso se va a presentar en pantalla.

Salida de vehículos

Cuando un vehículo va a salir del parqueo, el cliente debe presentar el ticket que le fue entregado en el ingreso. El usuario del sistema ingresa el número de placa y el sistema automáticamente calcula el tiempo que el vehículo estuvo en el estacionamiento y realiza el cálculo del costo total del servicio. El cálculo de la tarifa debe hacerse en múltiplos de 15 minutos. Por ejemplo, si el tiempo en el estacionamiento fue 1 hora y 5 minutos, debe cobrarse 1 hora y 15 minutos. Si el tiempo fue de 47 minutos, se cobra 1 hora entera. Siempre se cobra hacia el siguiente múltiplo de 15 minutos.

El programa debe mostrar en pantalla el monto que debe pagar el cliente y mostrar una opción para realizar el cobro.

Cobro

Una vez que el cliente conoce el monto del servicio, se procede a realizar el cobro. El programa debe facilitar un espacio para que el usuario escriba la cantidad de efectivo con la que el cliente está pagando. Por ejemplo, con un billete de 2000 o 5000. Una vez que se introduce este dato, el programa calcula el monto del vuelto que debe darse al cliente y procede a imprimir la factura. Para efectos de este proyecto la factura tampoco se va a imprimir, sino que solamente se presentará en pantalla. A continuación se muestra un posible ejemplo de una factura.

Parqueo Morazán

Abierto las 24 horas. Tel: 48657348
Cédula jurídica: 1-1111-1112
Dirección: Costado norte el Parque Morazán, San José

Fecha: 14/03/2014 Número: 000001

Total:

El número de las facturas es secuencial, por lo que nunca debe darse el caso en el que dos facturas tengan el mismo número.

Cerrar el parqueo

Esta opción se ejecutará cuando el parqueo termine sus servicios. Sólo se puede cerrar el parqueo si todos sus espacios se encuentran desocupados. Debe guardarse en el archivo XML el consecutivo de la siguiente factura, para que la próxima vez que se abra el parqueo se continúe con la misma numeración. También, el propósito de hacer el cierre es determinar cuánto dinero hay en la caja al final del día y cuánta es la diferencia con el monto que se registró al inicio del día, es decir, cuánto dinero se ganó durante el día. Debe mostrarse en pantalla la cantidad de dinero en caja y la ganancia.

Entradas y salidas de efectivo

Es posible que durante el día se necesite agregar efectivo extra a la caja o, por el contrario, sacar efectivo de la caja. Estas operaciones sólo pueden ser ejecutadas por el administrador, por lo cual deben solicitar la contraseña para poder ejecutarlas y preguntar cuánto es el monto que se va a ingresar o se va a retirar de la caja.

Ver estado de parqueo

Esta funcionalidad puede ser utilizada por el usuario en cualquier momento y debe mostrar los espacios del parqueo en alguna forma gráfica y dentro de los espacios mostrar cuáles tienen vehículos y su número de placa.

Consultar historial de facturas

Esta opción permite al usuario ver la lista de facturas que han sido entregadas desde que se abrió el parqueo, con el propósito de poder realizar consultas. Al seleccionar la factura deseada, debe mostrarse en pantalla de la misma forma que se hace durante el cobro.

Estructura

El programa debe presentar al usuario las siguientes opciones. La estructura no debe ser igual, puede variarse bajo el criterio del equipo para mejorar la usabilidad de la aplicación.

1. Abrir parqueo
Verifica que el parqueo tenga espacios definidos y que la caja tenga el monto mínimo (parámetro del sistema).
2. Ingreso de vehículo
Recibe el número de placa y otros datos del vehículo, registra la hora de entrada y el espacio asignado al vehículo, despliega el ticket que se entrega al usuario.
3. Salida de vehículo
Recibe el número de placa o ticket y marca la hora de salida del vehículo. Calcula el tiempo de servicio y el monto a pagar (múltiplos de 15 minutos). Muestra opción para generar factura.
 - a. Generar factura
Recibe la cantidad de efectivo entregada por el cliente, calcula el vuelto que debe entregarse, aumenta la cantidad de efectivo en caja y despliega la factura que se entrega al usuario.
4. Estado del parqueo
Muestra los espacios del parqueo, cuáles están ocupados y los números de placa de los vehículos que los están ocupando.
5. Historial de facturas
Muestra una lista de todas las facturas generadas desde que se inició la aplicación. Al seleccionar alguna, se muestra la factura de la misma forma que en el punto 3.a.
6. Caja
Muestra la cantidad de efectivo que hay en la caja actualmente.
 - a. Entrada de efectivo (requiere contraseña)
Debe corroborar que la contraseña introducida corresponde con la de administrador. Recibe un monto de dinero y aumenta en esa cantidad el efectivo de la caja.
 - b. Salida de efectivo (requiere contraseña)
Debe corroborar que la contraseña introducida corresponde con la de administrador. Recibe un monto de dinero y disminuye en esa cantidad el efectivo de la caja.
7. Cerrar parqueo
Comprueba que todos los espacios del parqueo estén vacíos. Muestra cuánto efectivo había en la caja al iniciar el programa, cuánto efectivo hay en la caja actualmente y la diferencia entre esas dos cantidades. Indica que ahora el parqueo está cerrado.
8. Configuración de parámetros (requiere contraseña)
Esta sección del programa permite cambiar los valores parametrizables del programa. Deben estar almacenados en un archivo XML. Si hay otros valores que se guarden en el archivo, también deben aparecer en esta sección.
 - a. Espacios del parqueo
Permite definir los espacios que tiene el parqueo. Cada espacio tiene un número (consecutivos) y una descripción breve.
 - i. Agregar espacio
 - ii. Eliminar espacio
 - b. Nombre
 - c. Dirección
 - d. Teléfono
 - e. Eslogan
 - f. Cédula jurídica

- g. Monto mínimo en caja para abrir
 - h. Cambiar contraseña de administrador (requiere contraseña)
9. Salir

La salida del programa debe permitirse solamente si la caja tiene un monto de cero. De otra forma la aplicación debe recordar al usuario que debe retirar todo el efectivo de caja antes de salir.

Interfaz

La aplicación debe contar con interfaz gráfica de usuario. No se evaluará el valor estético, si no su funcionalidad y usabilidad.

Documentación

Como documentación interna, los métodos deben llevar como comentarios las entradas que recibe, las salidas que entrega y las restricciones que tiene. Evite hacer comentarios excesivos. Escriba código claro y conciso, trate de apegarse a los principios de código limpio para el código que escriba. Recuerde que es mucho más importante la claridad de su código que su eficiencia.

En cuanto a la documentación externa, debe entregarse dos documentos en formato PDF:

1. Manual de usuario: sección que explica detalladamente a cualquier usuario cómo ejecutar y utilizar la aplicación. Debe tener portada.
2. Documentación del proyecto
 - a. Portada
 - b. Resumen ejecutivo. Descripción breve de todo lo que abarca la documentación. El objetivo de este resumen es captar la atención del lector y motivarlo a aprender más sobre el proyecto. Menos de una página.
 - c. Introducción. ¿Por qué se hace el proyecto y qué se incluye? (1-2 págs.)
 - d. Presentación y análisis del problema (5+ págs.)
 - i. Qué es lo que hay que resolver. Identificar pequeños problemas que deben resolverse en el proyecto.
 - ii. Cómo se va resolver el problema. La forma en que se planea resolver el problema.
 - iii. Análisis crítico de la implementación. Luego de la implementación, decir qué se logró implementar, lo que faltó y qué cosas se podrían mejorar de lo que se implementó. No sólo mencionar, si no explicar por qué.
 - e. Diagrama de clases detallado.
 - f. Conclusiones: resoluciones puntuales tras el proyecto. Estas deben ser relacionadas con los aspectos técnicos del trabajo únicamente. (1-2 págs.)
 - g. Recomendaciones: consejos o advertencias que se derivan de las conclusiones. Lecciones aprendidas durante el desarrollo del proyecto. Recomendaciones para personas que tengan que hacer el mismo trabajo. También deben estar orientadas con aspectos técnicos de la tarea programada. Se recomienda hacer una o más recomendaciones por cada conclusión. (1-2 págs.)
 - h. Referencias. Deben incluirse en formato APA.

Forma de trabajo

El proyecto se desarrollará en tríos o parejas. No se permitirá la entrega de proyectos desarrollados de forma individual a menos que exista autorización previa del profesor.

Se recomienda encarecidamente presentar al menos un adelanto de trabajo (diagrama de clases) al profesor o asistentes del curso durante la primera semana y media del proyecto. Esta supervisión le puede ayudar a mejorar el resultado final del proyecto. No dude en consultar cualquier asunto tanto de programación como de elaboración de la documentación.

Entrega

El tiempo asignado para la tarea programada es de 3 semanas. La fecha de entrega y medios de envío serán definidas por el profesor en el momento de la entrega de este enunciado.

Evaluación

La tarea tiene un valor de 15% de la nota final, en el rubro de Proyectos.

Desglose de la evaluación de la tarea programada:

Documentación: 35%

Programación: 65%

Recomendaciones adicionales

Pruebe cada funcionalidad individualmente. No implemente todo el programa sin verificar el funcionamiento por separado de cada una de sus partes. Esto dirige a errores que son más difíciles de encontrar.

Recuerde que el trabajo es en equipos, es indispensable la comunicación y la coordinación entre los miembros del subgrupo.

Comparta el conocimiento con los demás compañeros de grupo y de la carrera, la ciencia de la computación es una disciplina que requiere el traspaso libre de conocimientos. Se logran mejores resultados con la colaboración de todos que con el esfuerzo separado de diferentes personas.

No dude en consultar diferentes fuentes para satisfacer las dudas. Aparte de las búsquedas en internet, asegúrese de exponer sus dudas a sus compañeros, profesor y conocidos que estudien la carrera; en la mayoría de las ocasiones es más provechosa una conversación de 10 minutos entre personas que están trabajando en lo mismo que pasar horas buscando la respuesta a una duda de forma individual.

No deje la documentación para el final, es buena práctica ir desarrollándola durante todo el transcurso del proyecto. Recuerde que la documentación debe ser concisa y puntual, por lo que en realidad no toma mucho tiempo al realizarla de esta forma.

Plagios no serán tolerados bajo ninguna circunstancia. Cualquier intento de fraude será evaluado con una nota de cero y se enviará una carta al expediente del estudiante. Siempre escriba su propio código.