

Aplicacion de gestion para el parqueadero La Ceiba

Link del repositorio: [zacwastaken/la-ceiba-parking-lot: This is a Java app for managing the clients, parking spaces and income of La Ceiba parking lot. \(github.com\)](https://github.com/zacwastaken/la-ceiba-parking-lot)

Integrantes:

- Juan David Cruz Garcia A00368421 [1KVueltasAlCampo \(Juan David Cruz Garcia\) \(github.com\)](https://github.com/1KVueltasAlCampo)
- Juan Sebastian Perez Camacho A00369582 [Juanperezaa \(github.com\)](https://github.com/Juanperezaa)
- Samuel Hernandez Cifuentes A00368069 [zacwastaken \(Samuel Hernandez\) \(github.com\)](https://github.com/zacwastaken)

Para esta segunda entrega, están listos los requerimientos:

- **Req 1. Gestionar** los clientes del parqueadero. Los clientes cuentan con un nombre (incluidos los apellidos), un número de celular y un número de identificación que puede ser opcional.
- **Req 4. Gestionar** los usuarios del parqueadero. Los usuarios cuentan con un nombre (incluidos apellidos), un username, un número de identificación y una contraseña.

Si bien estos no se encuentran implementados en la interfaz gráfica, por medio de los testCase de JUnitCase5 se puede comprobar que todos los métodos relacionados a estos requerimientos son totalmente funcionales.

Requerimientos funcionales

El sistema debe estar en la capacidad de:

Req 1. Gestionar los clientes del parqueadero. Los clientes cuentan con un nombre (incluidos los apellidos), un número de celular y un número de identificación que puede ser opcional.

- Req 1.1 Agregar un nuevo cliente al parqueadero. Es decir registrar su respectiva información.

- Req 1.2 Deshabilitar un cliente del parqueadero. Esto solo debe ser posible si ningún vehículo tiene a este cliente como propietario.
- Req 1.3 Actualizar la información del cliente.
 - Req 1.3.1 Cambiar el nombre del cliente. No debe ser posible modificar este para que coincida con otro cliente en el parqueadero.
 - Req 1.3.2 Cambiar el numero de celular del cliente.
 - Req 1.3.3 Cambiar el número de identificación del cliente. No debe coincidir con la identificación de otro cliente.

Req 2. Gestionar los vehículos del parqueadero. Cada vehículo tiene un tipo (moto, carro, bus, camion, etc), un tiempo de estadía (uno o varios días, mensual, o indefinido), un modelo, una placa, un color, un propietario, un puesto en el parqueadero, un campo de observaciones opcional y un valor a pagar en base al tiempo de estadía definido y el tipo del vehículo. Si el valor a pagar con el tiempo de estadía por hora/diario excede el valor de la mensualidad entonces se cambiará el tipo de estadía a mensual con el fin de evitar un pago abusivo.

- Req 2.1 Añadir un vehículo al parqueadero con sus respectivas características
- Req 2.2 Deshabilitar un vehículo cuando este ya no se encuentre en el parqueadero.
- Req 2.3 Actualizar la información de un vehículo. No se puede cambiar ni el tipo ni el tiempo de estadía definido.
 - Req 2.3.1 Cambiar el modelo del vehículo.
 - Req 2.3.2 Cambiar la placa del vehículo. No debe ser posible modificarla para que coincida con la placa de otro vehículo en el parqueadero.
 - Req 2.3.3 Cambiar el color del vehículo
 - Req 2.3.4 Cambiar el puesto en el parqueadero del vehículo. No debe poder cambiarse por un puesto ocupado.
 - Req 2.3.5 Cambiar el campo de observaciones del vehículo

Req 3. Representar un mapa virtual del parqueadero en donde se puedan visualizar los vehículos que están en cada uno de los puestos del parqueadero. Existen 9 espacios para vehículos grandes, 20 para vehículos pequeños y 9 para motos.

- Req 3.1 Permitir que al dar click sobre el icono del vehículo en un puesto específico. Se muestra la información de este (color, modelo, placa, etc).
- Req 3.2 Permitir eliminar un vehículo. Se debe indicar si el propietario del vehículo pagó o no.

Req 4. Gestionar los usuarios del parqueadero. Los usuarios cuentan con un nombre (incluidos apellidos), un username, un número de identificación y una contraseña.

- Req 4.1 Añadir un usuario al parqueadero con su respectiva información.
- Req 4.2 Eliminar un usuario del parqueadero.
- Req 4.3 Actualizar la información de los usuarios. Para permitir cada cambio es necesario ingresar el nombre de usuario y la contraseña actual.
 - Req 4.3.1 Cambiar el nombre del usuario. No debe ser posible modificarlo para que coincida con el nombre de otro usuario del parqueadero.
 - Req 4.3.2 Cambiar el username. No debe ser posible modificarlo para que coincida con otro username del parqueadero.
 - Req 4.3.3 Cambiar el número de identificación. No debe ser posible modificarlo para que coincida con el número de identificación de otro usuario del parqueadero.
 - Req 4.3.4 Cambiar la contraseña.

Req 5. Clasificar la información de los vehículos de acuerdo a su tiempo de estadía.

- Req 5.1 Agrupar en una tabla la información de los vehículos cuyo tiempo de estadía es por hora/diario.

- Req 5.1.1 Asignar un número (en orden empezando desde 1) de recibo para cada vehículo.
- Req 5.1.2 Mostrar la fecha en que ingresó el vehículo.
- Req 5.1.3 Mostrar el tipo de vehículo.
- Req 5.1.4 Mostrar la placa del vehículo.
- Req 5.1.5 Mostrar la hora de ingreso del vehículo.
- Req 5.1.6 Mostrar la hora de salida del vehículo.
- Req 5.1.7 Mostrar el valor que este debe pagar. Si el cliente no ha pagado el color de esta casilla es rojo, si ya está pagado el color de esta casilla es verde.
- Req 5.1.8 Mostrar las observaciones del vehículo.
- Req 5.2 Agrupar en una tabla la información de los vehículos cuyo tiempo de estadía es mensual.
 - Req 5.2.1 Asignar un número (en orden empezando desde 1 para cada mes)
 - Req 5.2.1 Mostrar la placa del vehículo.
 - Req 5.2.2 Mostrar al propietario del vehículo.
 - Req 5.2.3 Mostrar el puesto en el parqueadero del vehículo.
 - Req 5.2.4 Mostrar la fecha del último pago del cliente.
 - Req 5.2.5 Mostrar la fecha en que el cliente debe pagar.
 - Req 5.2.6 Mostrar el valor que se debe pagar por el servicio. Si el cliente no ha pagado el color de esta casilla es rojo, si ya pagó el color de esta casilla es verde.

Req 6. Generar un recibo en caso de que un cliente lo solicite. Los recibos para vehículos cuyo tiempo de estadía es mensual cuentan con un número de recibo, la ciudad donde se

realiza el pago, el valor pagado (en número y en letras), la fecha en que se realiza el recibo, el nombre del cliente y un texto donde se indica por cual vehículo se pago y cuantos días estuvo. Los recibos para vehículos cuyo tiempo de estadía es por hora/diario cuentan con el tipo de vehículo, la fecha en que se realiza el recibo, el valor que se paga, la placa del vehículo, la fecha y hora de entrada del vehículo y el total de horas que estuvo el vehículo.

Req 7. Generar reportes correspondientes a la información del parqueadero en un rango de fechas definidas por el usuario.

- Req 7.1 Generar un reporte que muestre la información de los clientes.
- Req 7.2 Generar un reporte que muestre generalmente la información de los vehículos que han estado en el parqueadero hasta el momento.
- Req 7.3 Generar un reporte que muestre la información de los vehículos cuyo tiempo de estadía es por hora/diario haciendo énfasis en aquello relevante al pago.
- Req 7.4 Generar un reporte que muestre la información de los vehículos cuyo tiempo de estadía es mensual haciendo énfasis en aquello relevante al pago.

Req 8. Guardar toda la información del parqueadero de forma persistente. Es decir que al cerrar el programa, este debe conservar la información.

Req 9. Permitir importar información desde un archivo de excel para cada elemento del parqueadero.

Req 10. Permitir la renovación del pago de un cliente.

Req 11. Agrupar a los vehículos con deudas pendientes.

Req 12. Buscar por medio de búsqueda binaria a los clientes, vehículos y usuarios del parqueadero.

- Req 12.1 Buscar por nombre a un cliente.
- Req 12.2 Buscar por placa a un vehículo.
- Req 12.3 Buscar por nombre a un usuario.

Req 13. Permitir importar información desde un archivo de texto plano para cada elemento del parqueadero.

Req 14. Registrar la información relacionada con el servicio de lavadero del parqueadero.

Req 15. Realizar un sistema de inicio de sesión con el fin de dar seguridad al programa. El inicio de sesión se hace en base a los usuarios anteriormente mencionados.

Requerimientos NO Funcionales

El sistema debe estar en la capacidad de:

Req 1. Modificar la información del programa únicamente cuando un usuario haya iniciado sesión.

Req 2. Realizar búsquedas binarias de forma eficiente procurando que sin importar la densidad de los datos estas se demoren más de 3.5 segundos.

Req 3. Operar adecuada y eficazmente con información que involucre hasta 2000 datos.

Req 4. Implementar métodos de ordenamiento efectivos.

Req 5. Procurar que la información importada sea recibida correctamente y sin errores en el proceso.

Req 6. Evitar el ingreso de datos repetidos.

Req 7 Manejar adecuadamente las posibles excepciones del programa.

Req 8. Poseer interfaces gráficas bien formadas.

Req 9. Ejecutarse adecuadamente en sistemas operativos Windows.

Req 10. Manejar adecuadamente la persistencia del programa procurando que no se pierda ningún dato.

Configuración de los Escenarios

Nombre	Clase	Escenario
vehicleSetupScenary1	Vehicle	Es el día 27 de mayo de 2021.
vehicleSetupScenary2	Vehicle	Nada

LargeVehicleSetupScenary1	LargeVehicle	Es el día 27 de mayo de 2021.
LargeVehicleSetupScenary2	LargeVehicle	Nada
parkingLotScenary1	ParkingLot	El parqueadero se crea y crea un cliente genérico.
parkingLotScenary2	ParkingLot	El parqueadero se crea y crea un empleado genérico.

Diseño de Casos de Prueba

Vehículo genérico

Objetivo de la Prueba: Verificar el correcto cálculo de la fecha en que el vehículo debería salir del parqueadero según el tiempo de estadía que se definió.

Clase	Método	Escenario	Valores de Entrada	Resultado
Vehicle	calculateExitDate	vehicleSetupScenary1	stayIndicator = 2 (Mes) numberOfTime = 1	Se calcula correctamente la fecha de salida del vehículo
Vehicle	calculateExitDate	vehicleSetupScenary2	stayIndicator = 0 (Hora) numberOfTime = 3	Se calcula correctamente la fecha de salida del vehículo
Vehicle	calculateExitDate	vehicleSetupScenary3	stayIndicator = 1 (Día) numberOfTime = 5	Se calcula correctamente la fecha de salida del vehículo
Vehicle	calculateExitDate	vehicleSetupScenary4	stayIndicator=3 (indefinido)	Se indica que no se puede calcular puesto que el tiempo de estadía es indefinido.

Objetivo de la Prueba: Verificar si al llamar al método takeOut se “da salida” a un vehículo del parqueadero.

Clase	Método	Escenario	Valores de Entrada	Resultado
Vehicle	takeOutTest	vehicleSetupScenary2	Ninguno	Se confirma que el vehículo ya no se encuentra en el parqueadero

Objetivo de la Prueba: Verificar si el vehículo permanece/permaneció en el parqueadero más tiempo de lo esperado

Clase	Método	Escenario	Valores de Entrada	Resultado
-------	--------	-----------	--------------------	-----------

Vehicle	checkAdditionalTimeTest1	vehicleSetupScenary1	Ninguno	Se indica que el vehículo estuvo más tiempo en el parqueadero de lo que acordó
Vehicle	checkAdditionalTimeTest2	vehicleSetupScenary2	Ninguno	Se indica que el vehículo NO ha estado más tiempo en el parqueadero de lo que acordó
Vehicle	checkAdditionalTimeTest3	vehicleSetupScenary2	Ninguno	Se indica que el vehículo NO ha estado más tiempo en el parqueadero de lo que se acordó porque en primer lugar no se acordó un tiempo definido

Vehículo grande

Objetivo de la Prueba: Verificar que los valores a pagar que el programa calcula para los vehículos grandes son los adecuados

Clase	Método	Escenario	Valores de Entrada	Resultado
LargeVehicle	calculateValueToPayTest1	LargeVehicleSetupScenary2	vehicleType = 2 (camión) stayIndicator = 0 (Hora) numberOfTime = 1	Se comprueba que el vehículo debe pagar 3000 pesos por 1 hora
LargeVehicle	calculateValueToPayTest2	LargeVehicleSetupScenary2	vehicleType = 2 (camión) stayIndicator = 0 (Hora) numberOfTime = 2	Se comprueba que el vehículo debe pagar 4000 pesos por 2 horas
LargeVehicle	calculateValueToPayTest3	LargeVehicleSetupScenary2	vehicleType = 2 (camión) stayIndicator = 0 (Hora) numberOfTime = 5	Se comprueba que el vehículo debe pagar 7000 pesos por 5 horas
LargeVehicle	calculateValueToPayTest4	LargeVehicleSetupScenary2	vehicleType = 2 (camión) stayIndicator = 0 (Hora) numberOfTime = 11	Se comprueba que el vehículo debe pagar 12000 pesos (Que es valor de un día puesto que el valor por horas sobrepasa este valor)
LargeVehicle	calculateValueToPayTest5	LargeVehicleSetupScenary1	vehicleType = 2 (camión) stayIndicator = 0 (Hora) numberOfTime = 2	Se comprueba que el valor a pagar es diferente a 4000 (valor de 2 horas) porque el vehículo estuvo más tiempo del que acordó. De igual forma se comprueba que se calcula adecuadamente el valor a pagar cuando el tiempo de estadía es indefinido.
LargeVehicle	calculateValueToPayTest6	LargeVehicleSetupScenary1	vehicleType = 3 (Bus) stayIndicator = 1 (Día) numberOfTime = 2	Se comprueba que el vehículo debe pagar 24000 pesos por 2 días
LargeVehicle	calculateValueToPayTest7	LargeVehicleSetupScenary1	vehicleType = 2 (Bus) stayIndicator = 2 (Hora) numberOfTime = 2	Se comprueba que el vehículo debe pagar 120000 pesos por 2 meses

Cientes

Objetivo de la Prueba: Verificar si al llamar al método addClient un cliente es agregado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	addClientTest	parkingLotScenary1	Ninguno	Se añade el cliente

Objetivo de la Prueba:. Verificar si al llamar al método disableClientByName un cliente es deshabilitado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	disableClientByNameTest	parkingLotScenary1	Ninguno	Se deshabilita el cliente

Objetivo de la Prueba:. Verificar si al llamar al método disableClientByID un cliente es deshabilitado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	disableClientByIDTest	parkingLotScenary1	Ninguno	Se deshabilita el cliente

Objetivo de la Prueba:. Verificar si al llamar al método updateClientNameTest el nombre del cliente es actualizado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	updateClientNameTest	parkingLotScenary1	Ninguno	Se actualiza el nombre del cliente

Objetivo de la Prueba:. Verificar si al llamar al método updateClientIDTest el ID del cliente es actualizado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	updateClientIDTest	parkingLotScenary1	Ninguno	Se actualiza el ID del cliente

Objetivo de la Prueba:. Verificar si al llamar al método updateClientCellNumber el Cell Number del cliente es actualizado

Clase	Método	Escenario	Valores de Entrada	Resultado
-------	--------	-----------	--------------------	-----------

ParkingLotTest	updateClientCellNumberTest	parkingLotScenary1	Ninguno	Se actualiza el Cell Number del cliente
----------------	----------------------------	--------------------	---------	---

Objetivo de la Prueba:. Verificar si al llamar al método clientVerifier encuentra atributos ya usados por otros clientes

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	clientVerifierTest	parkingLotScenary1	Ninguno	Se comprueba si algún atributo del nuevo cliente es usado por otro cliente.

Objetivo de la Prueba:. Verificar si al llamar al método searchByID devuelve un cliente.

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	SearchByIDTest	parkingLotScenary1	Ninguno	Devuelve un cliente el cual cuenta con dicho ID

Objetivo de la Prueba:. Verificar si al llamar al método searchByName devuelve un cliente.

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	SearchByNameTest	parkingLotScenary1	Ninguno	Devuelve un cliente el cual cuenta con dicho nombre

Empleado

Objetivo de la Prueba:. Verificar si al llamar al método addEmployee un cliente es agregado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	addEmployeeTest	parkingLotScenary2	Ninguno	Se añade el empleado

Objetivo de la Prueba:. Verificar si al llamar al método disableEmployeeByName un empleado es deshabilitado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	disableEmployeeByNameTest	parkingLotScenary2	Ninguno	Se deshabilita el empleado

Objetivo de la Prueba:. Verificar si al llamar al método disableEmployeeById un empleado es deshabilitado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	disableEmployeeByIdTest	parkingLotScenary2	Ninguno	Se deshabilita el empleado

Objetivo de la Prueba:. Verificar si al llamar al método updateEmployeeNameTest el nombre del empleado es actualizado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	updateEmployeeNameTest	parkingLotScenary2	Ninguno	Se actualiza el nombre del empleado

Objetivo de la Prueba:. Verificar si al llamar al método updateEmployeeIDTest el ID del empleado es actualizado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	updateEmployeeIDTest	parkingLotScenary2	Ninguno	Se actualiza el ID del empleado

Objetivo de la Prueba:. Verificar si al llamar al método updateEmployeeUsername el nombre del empleado es actualizado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	updateEmployeeUsernameTest	parkingLotScenary2	Ninguno	Se actualiza el username del empleado

Objetivo de la Prueba:. Verificar si al llamar al método updateEmployeePassword el nombre del empleado es actualizado

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	updateEmployeePasswordTest	parkingLotScenary2	Ninguno	Se actualiza el password del empleado

Objetivo de la Prueba:. Verificar si al llamar al método employeeVerifyer encuentra atributos ya usados por otros empleados

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	employeeVerifyerTest	parkingLotScenary2	Ninguno	Se comprueba si algún atributo del nuevo empleado es usado por otro empleado.

Objetivo de la Prueba:. Verificar si al llamar al método employeeVerifyerLogin encuentra si el usuario o contraseña están siendo usados por otros empleados

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	employeeVerifyerLoginTest	parkingLotScenary2	Ninguno	Se comprueba si el usuario y contraseña del nuevo empleado es usado por otro empleado.

Objetivo de la Prueba:. Verificar si al llamar al método searchEmployeeByID devuelve un empleado..

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	SearchEmployeeByIDTest	parkingLotScenary2	Ninguno	Devuelve un empleado el cual cuenta con dicho ID

Objetivo de la Prueba:. Verificar si al llamar al método searchEmployeeByName devuelve un empleado..

Clase	Método	Escenario	Valores de Entrada	Resultado
ParkingLotTest	SearchEmployeeByNameTest	parkingLotScenary2	Ninguno	Devuelve un empleado el cual cuenta con dicho nombre

