

Diseño Proyecto 3

Alejandro Hoyos Mogollón – 202215277

Luisa Gómez Orrego – 202222249

Édison Julián León – 202021373

Emisión de facturas en PDF:

Se empleó la biblioteca Apache PDFBox para llevar a cabo la implementación de este requisito. Además, se ha introducido una nueva clase denominada `GeneradorFacturas`. Esta clase recibe todos los parámetros necesarios para la generación de una factura y crea el correspondiente archivo PDF, almacenándolo en la carpeta "facturas" de la base de datos. Es relevante destacar que esta función se activa exclusivamente al realizar reservas.

Características de los vehículos

Se ha introducido un nuevo atributo en la clase `Sistema` denominado `tipoVehiculos`. Este atributo se configura como un `HashMap`, donde las claves representan los tipos de vehículos y los valores indican el monto de la prima asociado a cada tipo. Al inicio del sistema, este `HashMap` se inicializa con los datos provenientes de un archivo de texto en la base de datos llamado "TipoVehiculos.txt".

Dicho archivo contiene la información necesaria para establecer los valores de los distintos tipos de vehículos.

Adicionalmente, se ha agregado un nuevo atributo a la clase Auto, denominado "tipoVehiculo". Este ajuste permite implementar la nueva característica relacionada con la clasificación de los vehículos según su tipo.

Pagos con Tarjeta de Credito

Para integrar las pasarelas de pago, se ha introducido una nueva clase denominada "PasarelaPago". Esta clase sirve como base para el desarrollo de futuras pasarelas, se requiere que estas nuevas clases extiendan de PasarelaPago y se incluyan las nuevas funcionalidades y/o atributos en estas. La adición de una nueva pasarela se lleva a cabo mediante la actualización del archivo "pasarelas.txt", ubicado en la carpeta "conf", con el nombre "uniandes.dpoo.(nombre de la pasarela)",

Esta última modificación posibilita la incorporación automática de las pasarelas en la consola del empleado, sin necesidad de ajustes adicionales. Finalmente, para implementar las nuevas funcionalidades en la consola relacionadas con las nuevas pasarelas de pago, es necesario añadir dichas especificaciones en el código, específicamente en la sección correspondiente a dichos pagos (ConsolaEmpleado).

App para Cliente

Para implementar la aplicación con interfaz grafica para cliente, se crearon las siguientes clases que extienden de **JFrame**:

App_Cliente:

Esta clase contiene ejecuta la primera interfaz que permite seleccionar una de las dos opciones: Autenticar usuario o crear usuario nuevo.



AppC_Form_LoginUsuario:

Esta clase extiende de JFrame y contiene el form para la autenticación del usuario. Esta se encarga de recibir los parámetros (login y contraseña) y comunicarse con la clase principal Sistema (Que previamente realizó la carga de datos) para verificar que el usuario Cliente efectivamente existe.



AppC_Form_NuevoUsuario:

Esta clase extiende de JFrame y contiene el form que permite la creación de nuevos usuarios Cliente. Esta se encarga de recibir los parámetros y si todos son correctos, es decir, si son coherentes con la lógica del negocio (Numero de celular es un entero positivo, también lo es numero de tarjeta, etc) permite añadir un nuevo cliente y lo guarda en el archivo de la base de datos Usuarios.txt mediante la clase principal Sistema.

CREAR NUEVO USUARIO

Nombre

Número de celular

Nacionalidad

Login

Foto de cédula

Foto de licencia

Buscar Archivo

Buscar Foto Licencia

ID de licencia

País de expedición de licencia

Fecha de vencimiento de licencia 09/12/2023

Tipo de Licencia

Numero de Tarjeta de Credito / Debito

Fecha de vencimiento de tarjeta 09/12/2023

Codigo de seguridad de tarjeta

Password

Mostrar

Aceptar

Volver

AppC_ConsultasReservas:

Esta clase, que hereda de JFrame, alberga el formulario destinado a la búsqueda de vehículos disponibles. Dicha búsqueda se lleva a cabo al verificar los parámetros de fecha inicial, fecha final y tipo de vehículo. Es crucial destacar que el acceso a este formulario está

condicionado a la autenticación previa del usuario como cliente, siendo necesario que el usuario cliente esté registrado en la base de datos.

Las fechas proporcionadas deben ser lógicamente coherentes con las reglas del negocio, es decir, deben ser mayores a la fecha actual, y la fecha final debe ser menor o igual a la fecha final de la reserva. Una vez que se han definido estos parámetros, la búsqueda de disponibilidad se inicia al hacer clic en el botón "Buscar Disponibilidad". Este proceso verifica la existencia de autos disponibles para las especificaciones proporcionadas.



The screenshot shows a Java Swing window titled "CONSULTA Y RESERVA USUARIO". It contains a form with the following fields and controls:

- Sede:** A dropdown menu with "Sede_Bogota" selected.
- Tipo Vehiculo:** A dropdown menu with "Automovil" selected.
- Prima adicional:** A text input field containing "\$10".
- Fecha de inicio:** A date picker showing "09/12/2023".
- Fecha final:** A date picker showing "09/12/2023".
- Buttons:** A "Buscar Disponibilidad" button is centered below the form. At the bottom, there are "Reservar" and "Cancelar" buttons.
- Display Area:** A large empty rectangular box is located below the "Buscar Disponibilidad" button, intended for displaying search results.

AppC_Reservas:

Esta clase extiende de JFrame y contiene el form y está destinada a la realización de reservas. Una vez se encuentra un vehículo de interés disponible, se procede a la realización de la reserva, Teniendo en cuenta el seguro del vehículo, y las tarifas (Tanto de categoría, como de tipo de vehículo en caso de que tenga una prima adicional) se realiza la reserva. En su etapa final, pregunta si desea realizar el pago automáticamente, o después (Teniendo en cuenta el descuento aplicado por hacerlo).

CONSULTA Y RESERVA USUARIO

Sede	Sede_Bogota
Tipo Vehiculo	Automovil
Prima adicional	\$10
Fecha de inicio	09/12/2023
Fecha final	09/12/2023

Buscar Disponibilidad

Categoria: van | Tipo Vehiculo: Automovil | Placa: SHY-000 | Marc

Reservar Cancelar

RESERVAR AUTO

Sede Inicio	Sede_Bogota
Sede Final	Sede_Bogota
Fecha Inicial	2023-12-09 23:27:17
Fecha Inicial	2023-12-09 23:27:17
Prima Adicional por riesgo	10
Seguro	Ninguno-\$0
Tarifa Categoría de vehiculo	1000
Recargo por entrega en otra sede	0

Aceptar Volver

RESERVAR AUTO

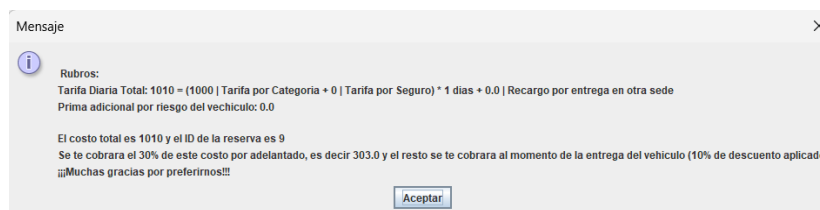
Sede Inicio	Sede_Bogota
Sede Final	Sede_Bogota
Fecha Inicial	2023-12-09 23:27:17

Confirmación

El costo total es de 1010
¿Desea realizar el pago del 30%?, es decir \$303.0 , esta acción te dará un 10% de descuento en el costo total del alquiler

Si No

Aceptar Volver



Pruebas automáticas:

Para la implementación de las pruebas se hizo uso de la librería Junit4. Se creó un nuevo paquete llamado test en donde se incluyen dos nuevas clases para cubrir los requerimientos.

testCargaArchivos:

Esta clase aborda la implementación de diversos escenarios tanto exitosos como de error durante el proceso de carga de archivos, involucrando todos los datos asociados a los alquileres, autos, pagos, sedes, seguros, tarifas y usuarios en la base de datos.

testIntegridadReservas:

Esta clase lleva a cabo pruebas que abarcan tanto escenarios exitosos como escenarios fallidos. Dado que la realización de reservas se efectúa tanto en la consola del cliente como en la aplicación del cliente, se ha incorporado un método denominado hacerReserva. Este método simula el mismo proceso utilizado en la consola y la aplicación, asumiendo, no obstante, que los parámetros ya han sido recibidos.

La implementación de estos parámetros se realiza a través del escenario de prueba por defecto, el cual incluye datos como el cliente, la categoría, las sedes, fechas, entre otros. Este enfoque permite evaluar la funcionalidad de manera integral, contemplando tanto situaciones exitosas como aquellas que podrían resultar en fallos.