**Documentación Proyecto 3**

Al principio encontrara las instrucciones de uso, mas abajo encontrara la documentacion y UML.

# **Emisión de facturas**

## **Paso 1:**

Al iniciar la aplicación aparecerá esta pantalla:

A screenshot of a computer

Description automatically generated

Por favor seleccione Iniciar como Cliente

## **Paso 2:**

A continuación llene las casillas de texto con Username: jj.diazo1 y Password: amoDpo y presione iniciar Sesión.

A screenshot of a computer

Description automatically generated

## **Paso 3:**

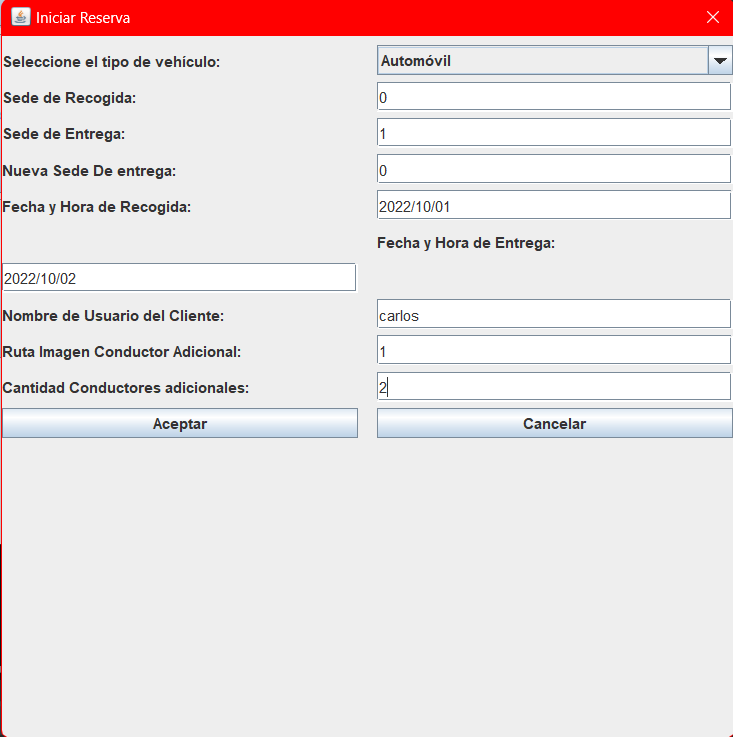
Para iniciar una reserva presione iniciar reserva:

A screenshot of a computer

Description automatically generated

## **Paso 4:**

Las 2 sedes que están habilitadas son la sede “0” y la sede “1” (escogimos estos nombres para facilidad de las pruebas, ya que era extenuante poner sedeMedellin cada vez que queriamos probarlo), por lo que es importante que escoja una de estas 2 sedes en el campo Sede de Recogida. Por otra parte note que el formato de la fecha es YY/MM/DD esto quiere decir que si quiere que la fecha de entrega sea el primero de Enero de 2022 se debe digitar así 2022/01/01.



Una vez termine de digitar los valores presione Aceptar.

## **Paso 5:**

Selecione uno de los 2 metodos de pago

## **A screenshot of a computer Description automatically generated**

## **Paso 6:**

Escriba el numero de su tarjeta. Para este paso incluimos 3 cuentas bancarias.

10: Tiene dinero y no debería tener problemas así que se realiza el pago:

A screenshot of a computer

Description automatically generated

20: No tiene dinero no se puede realizar la transacción.

A screenshot of a computer

Description automatically generated

30: Tiene la Cuenta bloqueada por lo que no se puede realizar el pago.

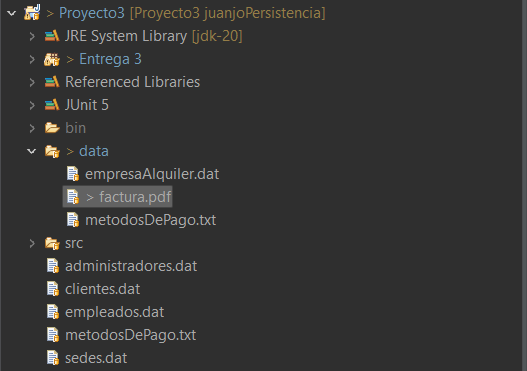
A screenshot of a computer

Description automatically generated

## **Paso 6:**

Una vez el pago haya sido realizado el recibo en PDF quedara guardado en la carpeta data con la información ingresada.

Ubicación:



Ejemplo de recibo:

A screenshot of a document

Description automatically generated

# **Verificar Disponibilidad de Vehículos**

## **Paso 1:**

Al iniciar la aplicación aparecerá esta pantalla:

A screenshot of a computer

Description automatically generated

Seleccione Iniciar como Cliente

## **Paso 2:**

A continuación llene las casillas de texto con Username: jj.diazo1 y Password: amoDpo y presione iniciar Sesión.

A screenshot of a computer

Description automatically generated

## **Paso 3:**

Para ver la disponibilidad de vehículos presione Ver Disponibilidad de Vehículos:

A screenshot of a computer

Description automatically generated

## **Paso 4:**

Las 2 sedes que están habilitadas son la sede “0” y la sede “1” por lo que es importante que escoja una de estas 2 sedes recomendamos usar sede 1 que es la que tiene la mayoría de los vehículos y de todos los tipo que se pedían para esta entrega. Por otra parte, note que el formato de la fecha es YY/MM/DD esto quiere decir que si quiere que la fecha de entrega sea el primero de Enero de 2022 se debe digitar así 2022/01/01.

A screenshot of a computer

Description automatically generated

## **Paso 5:**

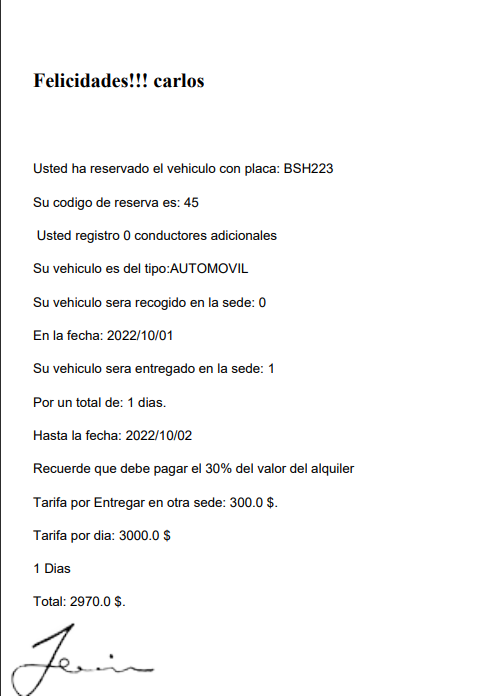
En esta ventana podrá ver todos los vehículos que están disponibles en la sede y el rango de fechas que usted ingreso para explorar la lista presione las flechas. Si así lo desea podrá iniciar reserva desde esta ventana y se le aplicara un descuento del 10 por ciento como se pidió en la guía. Para hacer la reserva siga los pasos desde el paso 4 de Emisión de Facturas.

A screenshot of a login box

Description automatically generated

## **Paso 6:**

Una vez terminada la reserva podrá observar que el precio de alquiler tuvo un descuento del 10 porciento ingresando los mismos datos.



# **3.Crear nuevo Cliente**

## **Paso 1:**

Al iniciar la aplicación aparecerá esta pantalla:

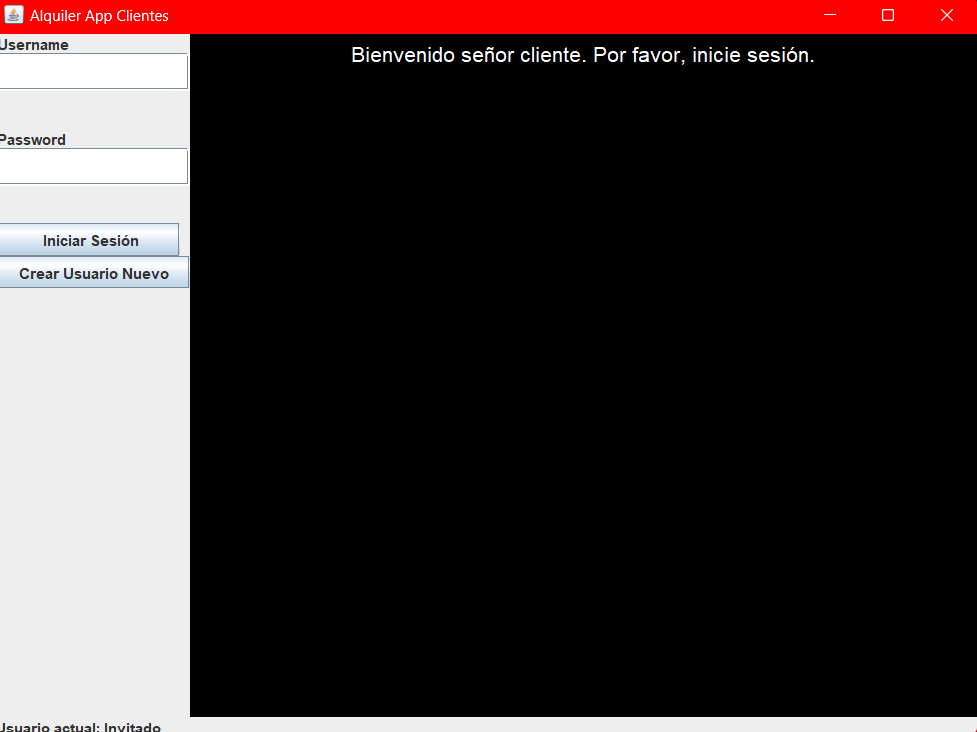
A screenshot of a computer

Description automatically generated

Seleccione Iniciar como Cliente

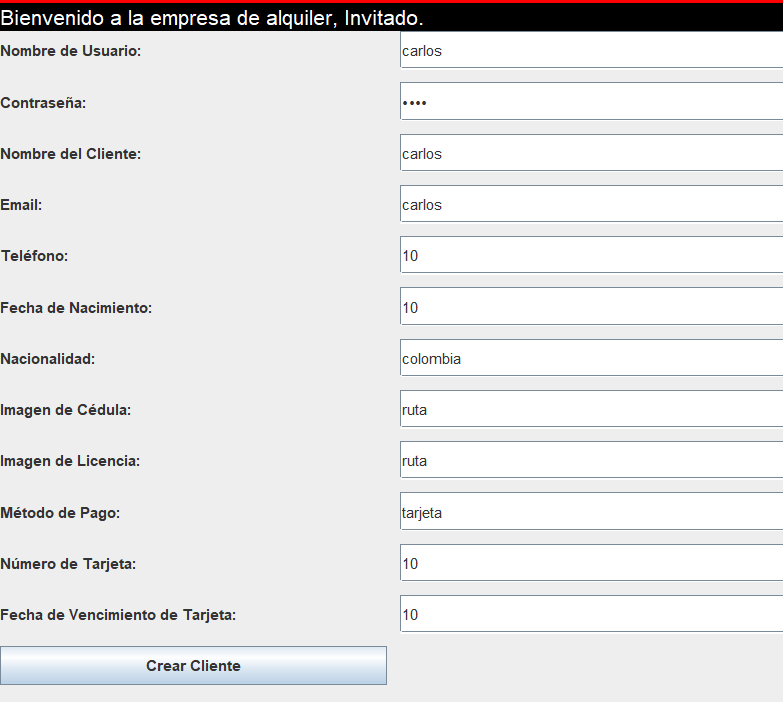
## **Paso 2:**

Seleccione Crear Usuario Nuevo



## **Paso 3:**

Ingrese la información de los clientes



Presione Crear Cliente

## **Paso 4:**

Ahora puede ingresar a la aplicación con el nuevo usuario que acaba de crear con el Username y el Password

A screenshot of a computer

Description automatically generated

## **Paso 5:**

Una vez ingresado el Username y el Password tendrá acceso al menú de clientes.

A screenshot of a computer

Description automatically generated

# **4.Carga dinámica:**

## **Paso 1**

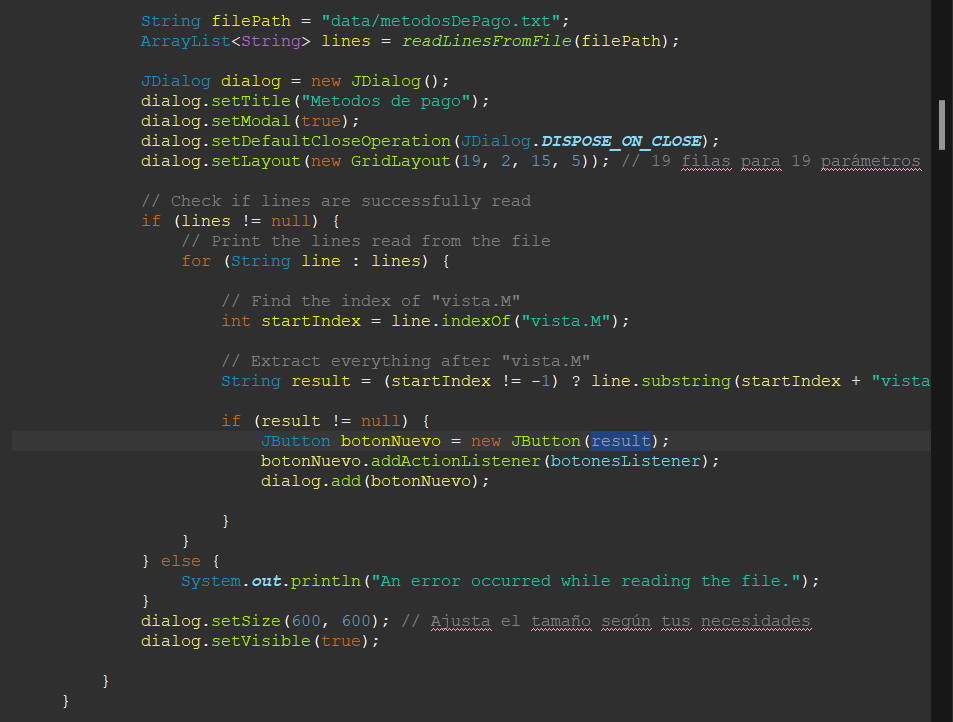
Lee el archivo de texto MetodosDePago y añade un botón por cada clase que haya dentro del archivo de texto.

A screenshot of a computer

Description automatically generated

A black background with white text

Description automatically generated



## **Paso 2**

Se detecta el botón que se presionó y se inicia la carga dinámica con la clase del botón asociado

A screen shot of a computer program

Description automatically generated

## **Paso 3**

Se busca la clase por nombre se ingresan los parámetros necesarios y se crea una instancia de la clase en cuestión.

# **Generalidades**

## **Cambios:**

## -Reservas y vehículos: La mayoría de cambios fue en Reserva, donde implementamos la nueva factura y el PDF. Introdujimos archivos de vehículos para poder probar ya que en entregas anteriores no habían vehículos.

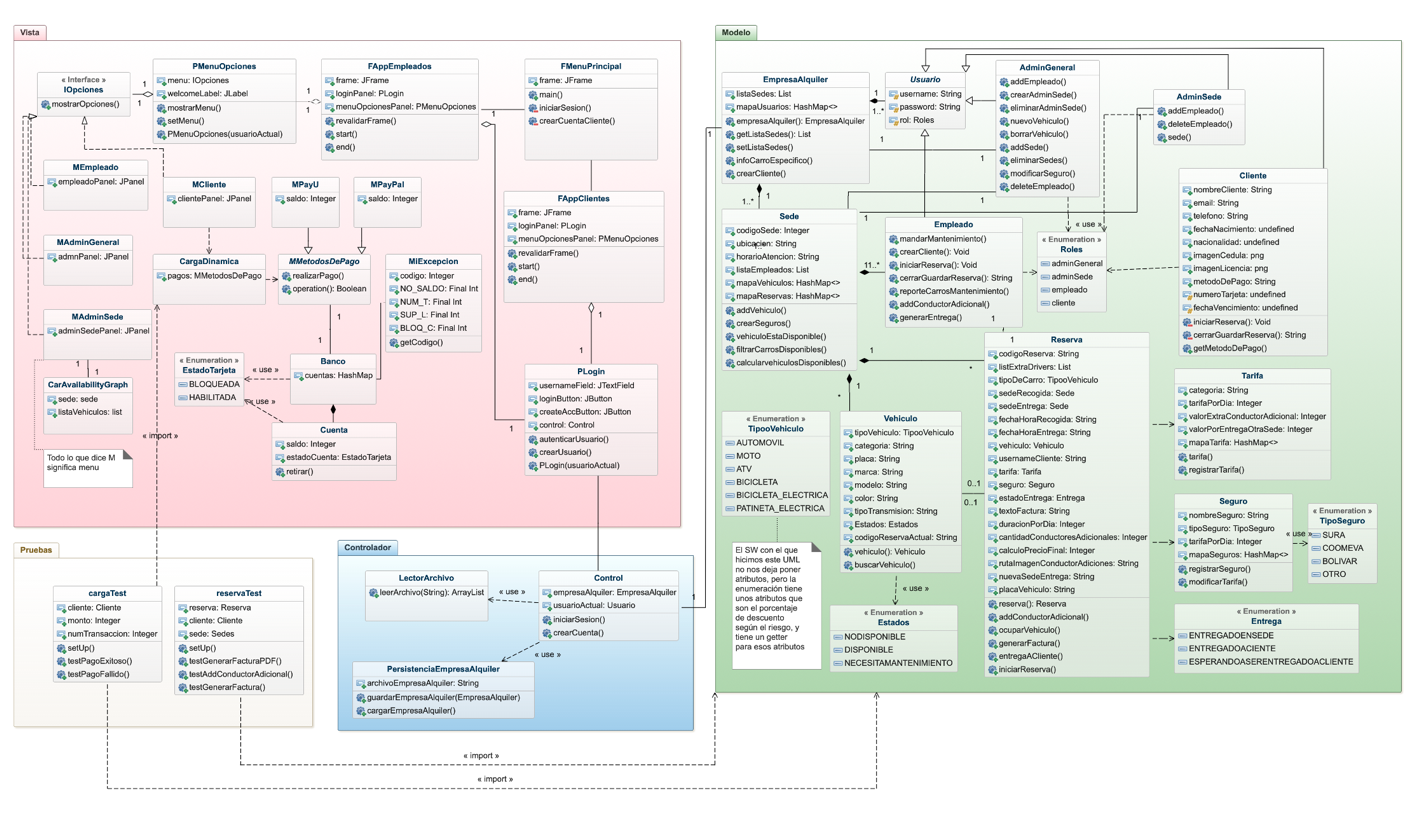
## -Separación de menús: Tenemos aparte de las sedes introducidas en el archivo sedes.dat, otras 2 más cuyo nombre son 0 y 1 por facilidad de pruebas, la sede 0 tiene pocos carros y la sede uno vendría siendo la sucursal, con más carros y empleados.

## -Nuevas clases en View: Tenemos 5 nuevas clases, CargaDinamica, Banco, MetodosDePago, MPayU y MPayPal, las cuales decidimos incluir en el view porque 3 de ellas son componentes visuales y las otras dos no tienen nada que ver con la lógica del negocio ya que están conectadas a entidades externas, en el caso de la carga dinámica tiene sus archivos correspondientes y en el caso del Banco, la idea es que fuera una entidad autónoma que ni si quiera estuviese dentro del proyecto si no solo un import, pero pusimos la clase con un mapa de cuentas para poder simular las transacciones en las pruebas.

## **Estructura:**

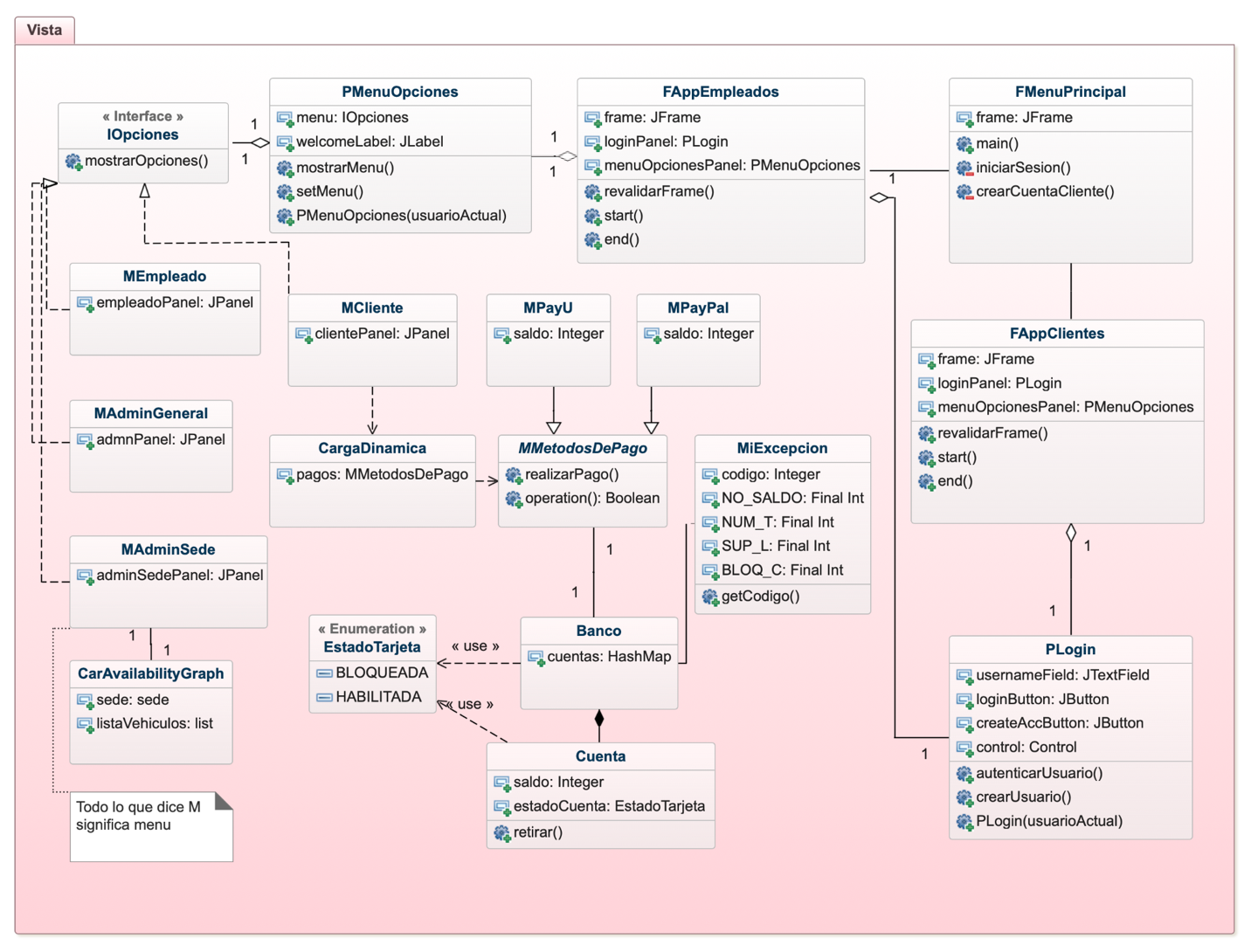
El proyecto consiste de 4 paquetes, lógica, control, vista, pruebas. Para esta entrega 3 hemos priorizado en optimizar el sistema de reservas, para ello separamos la aplicación de cliente y empleado cómo se pidió, por medio de dos frames distintos en el paquete vista, los cuales trabajan de la misma manera, la única diferencia es que al instanciar al PMenuOpciones y al PLogin (los dos paneles que van en la mitad del menú) se les introduce un entero cómo parámetro el cual con 0 ejecuta el menú para empleados y con 1 el menú para clientes, esta decisión la tomamos para no tener que modificar el código fuente de manera excesiva al separar los dos menús.

## **UML:**



# Este UML presenta de manera general el modelo de la aplicación, pero este podría no ser fiel del todo en cuanto a los atributos y métodos de las clases ya que al ser tantos, decidimos poner solo los mas importantes, enfocándonos mas en como interactúan las clases entre si y que relaciones tienen.

A continuación, presentamos un diagrama mas completo de la vista, que fue nuestro eje de trabajo principal para este proyecto:



Aca se puede apreciar de manera mas clara como hicimos los cambios mas importantes de la vista. Si quiere hacer un recorrido guiado sobre como funciona la aplicación, empiece desde la parte de arriba a la derecha, hacia la izquierda va a los menus y hacia abajo a las clases mas logicas, por ejemplo PLogin, quien es la encargada de manejar el acceso del usuario y tiene relacion con control, para acceder a los metodos de la logica de la aplicación.

## **PRUEBAS:**

Tuvimos problemas para realizar las pruebas de reserva, ya que al tener una estructura muy acoplada, fue duro de implementar las pruebas y al final no pudimos hacer que funcionaran. Pero las pruebas de la carga dinamica cubrieron el 86% y funcionaron perfecto, la razon por la que no cubrieron el 100% es porque tomamos la decision de que no era relevante probar un catch que ya se estaba manejando bien en la propia clase.

## **FIN**

Muchas gracias por usar nuestra aplicación.