

Fundamentos Ingeniería de Software

Entrega Proyecto: Documento Arquitectura

Santiago Guerrero, Santiago Barajas, Sebastián Isaza

2023

**Manejo de objetivos de arquitectura**

1. **Usabilidad**: Se debe asegurar que la aplicación sea fácil de usar y que la experiencia del usuario sea agradable. Esto podría incluir objetivos específicos como la implementación de una interfaz de usuario intuitiva y fácil de navegar, la optimización del rendimiento de la aplicación para una carga rápida y la incorporación de comentarios de los usuarios para identificar áreas de mejora.
2. **Disponibilidad**: La disponibilidad de la aplicación. Se debe garantizar que la aplicación esté disponible para los usuarios en todo momento y que los tiempos de inactividad se minimicen. Esto podría incluir objetivos específicos como la implementación de redundancia en la arquitectura para garantizar la continuidad del servicio en caso de fallos, la implementación de mecanismos de recuperación ante desastres y la realización de pruebas regulares de disponibilidad.
3. **Escalabilidad**: Se debe garantizar que la aplicación sea capaz de manejar cargas de trabajo más grandes a medida que el número de usuarios aumenta. Esto podría incluir objetivos específicos como la optimización del rendimiento de la base de datos y la implementación de técnicas de caché para reducir el tiempo de respuesta de la aplicación.
4. **Portabilidad**: Se debe garantizar que la aplicación pueda ser desplegada en diferentes plataformas, sistemas operativos y entornos de nube. Esto podría incluir objetivos específicos como la implementación de contenedores y la integración con proveedores de nube pública como AWS o Google Play Store.
5. **Seguridad**: Garantizar que la aplicación sea segura y proteja los datos del usuario de posibles amenazas. Esto podría incluir la implementación de medidas de autenticación y autorización robustas, la encriptación de datos sensibles y la realización de pruebas regulares de seguridad.
6. **Integración**: Garantizar la integración de la aplicación con otros sistemas y servicios relevantes para el negocio. Esto podría incluir la implementación de interfaces de programación de aplicaciones (API) para permitir la integración con sistemas de terceros y la integración con sistemas internos del negocio como el sistema de gestión de clientes o el sistema de facturación.

**Diagrama Caso de Uso (Vista Escenarios)**

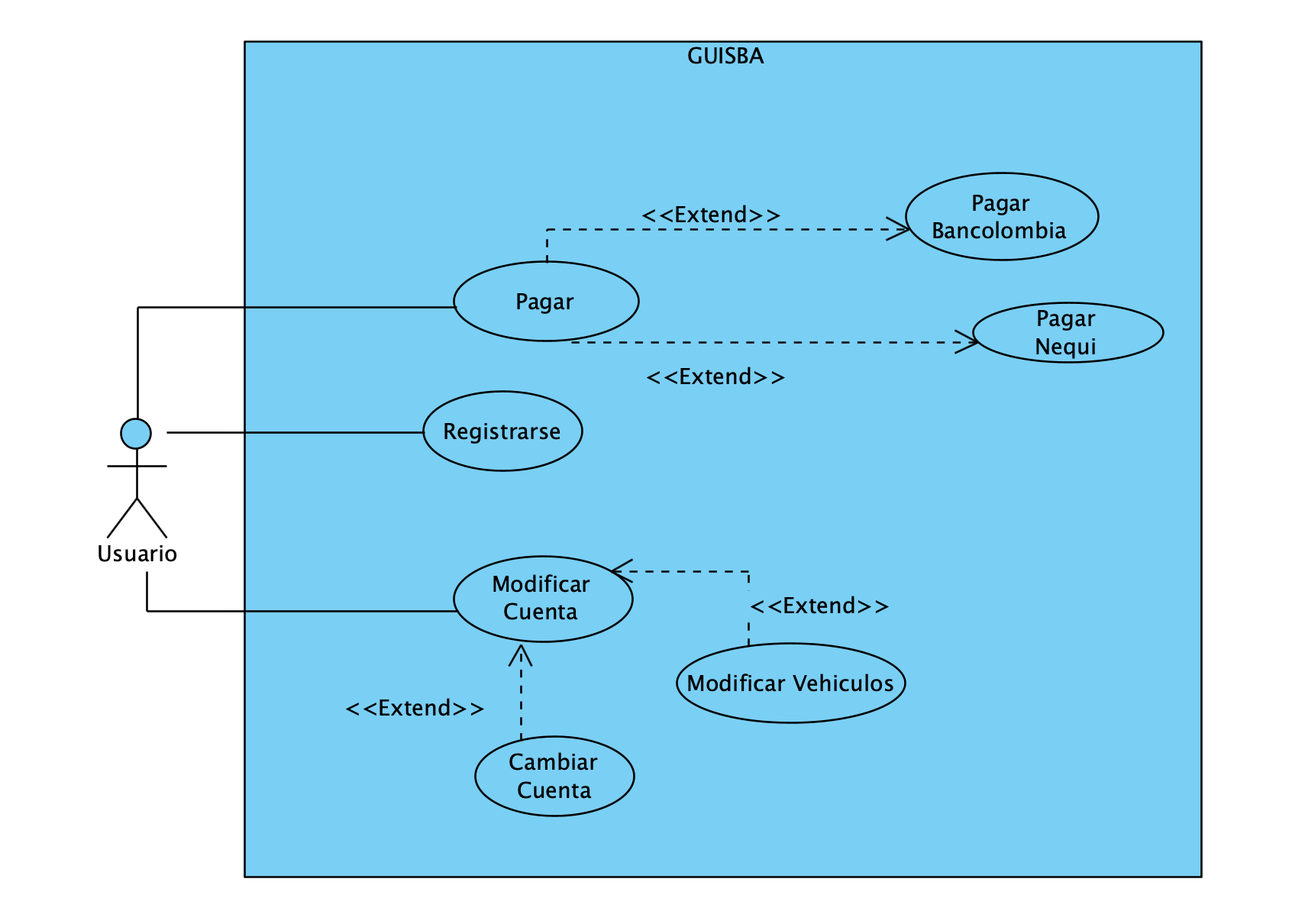
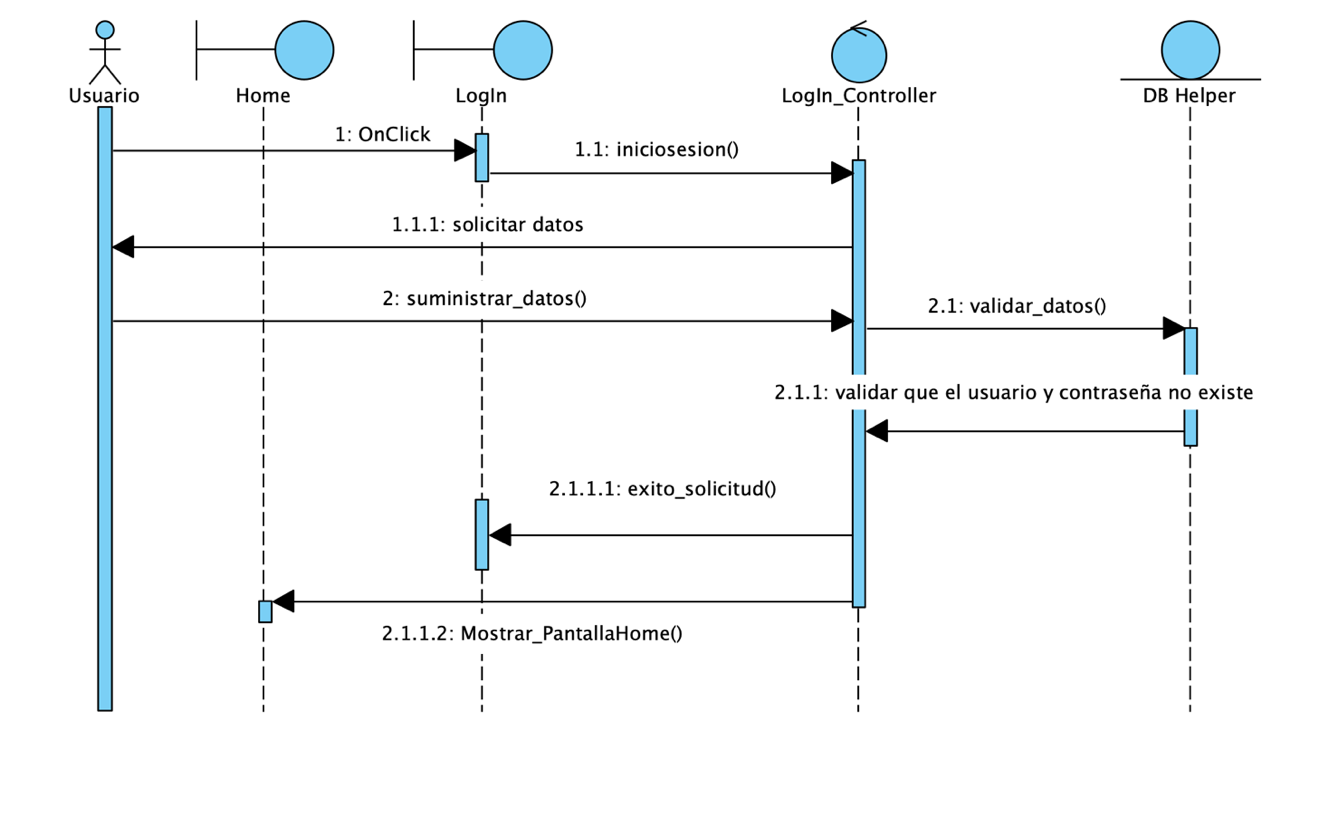


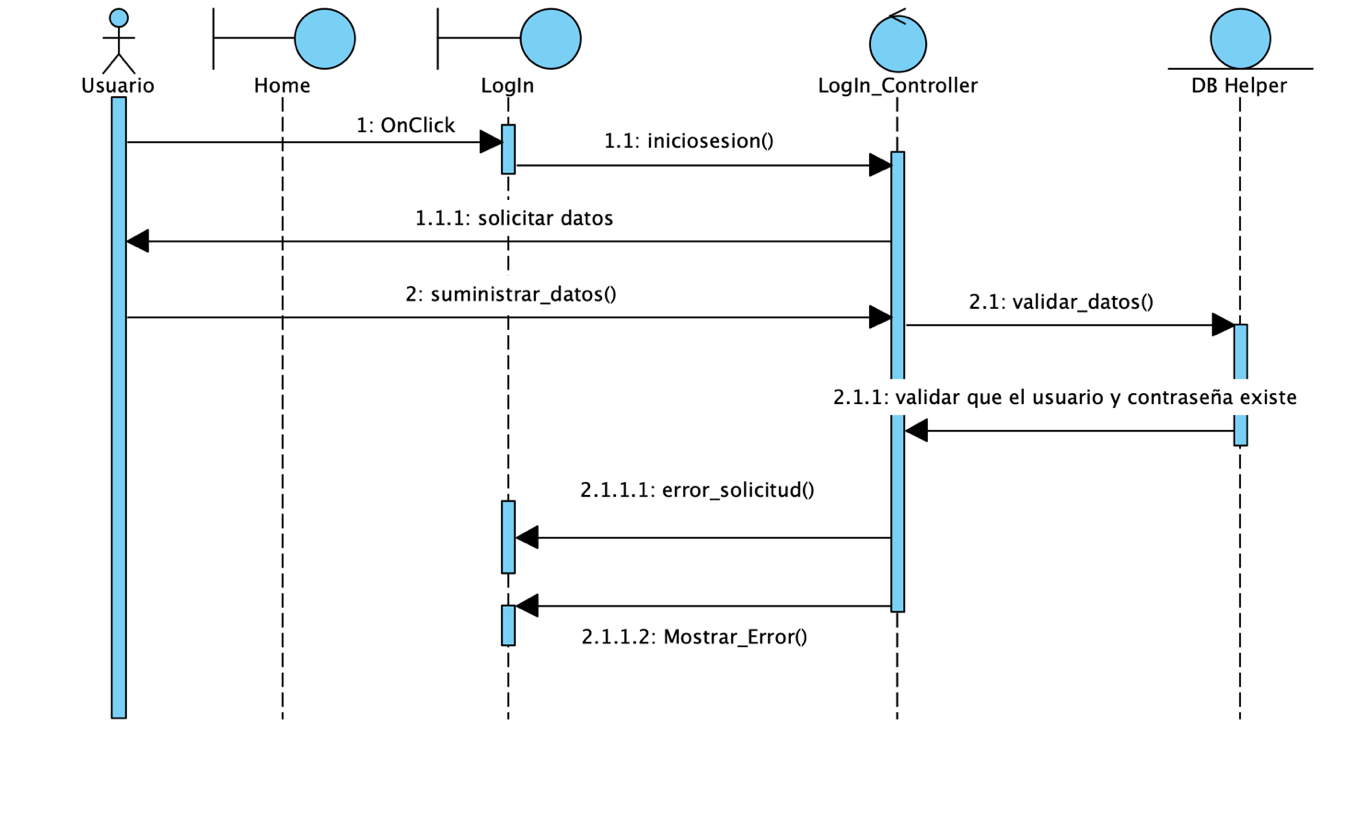
Diagrama de caso de uso en el que el actor principal es el usuario el cual se puede registrar en el sistema, para que tenga una cuanta y pueda realizar otras acciones como: pagar el parqueo del vehículo y modificar algún aspecto de su cuenta.

**Vista lógica**

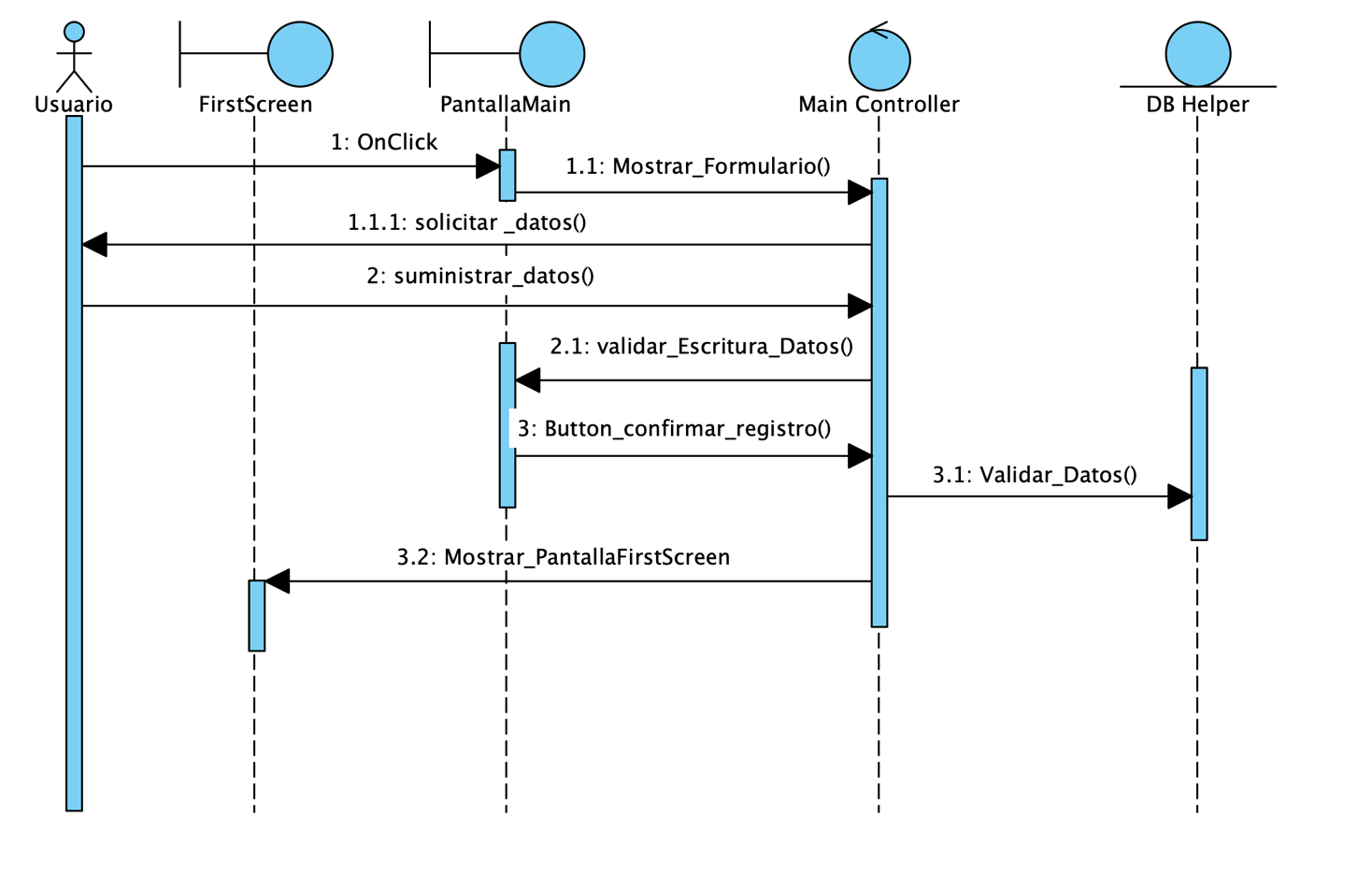
**Inicio de Sesión:** Diagrama de secuencia en el cual el usuario ingresa a la pantalla de Login y selecciona iniciar sesión, en la aplicación se cambia de pantalla y ahora procede a pedir los datos del usuario: correo y contraseña, posterior a esto el controlador del Login manda los datos a validar en la bade de datos “DB Helper” y si el usuario existe le manda confirmación al controller y este permite la entrada del usuario a la siguiente pantalla, en caso contrario se le notifica un error al usuario de credenciales invalidas.



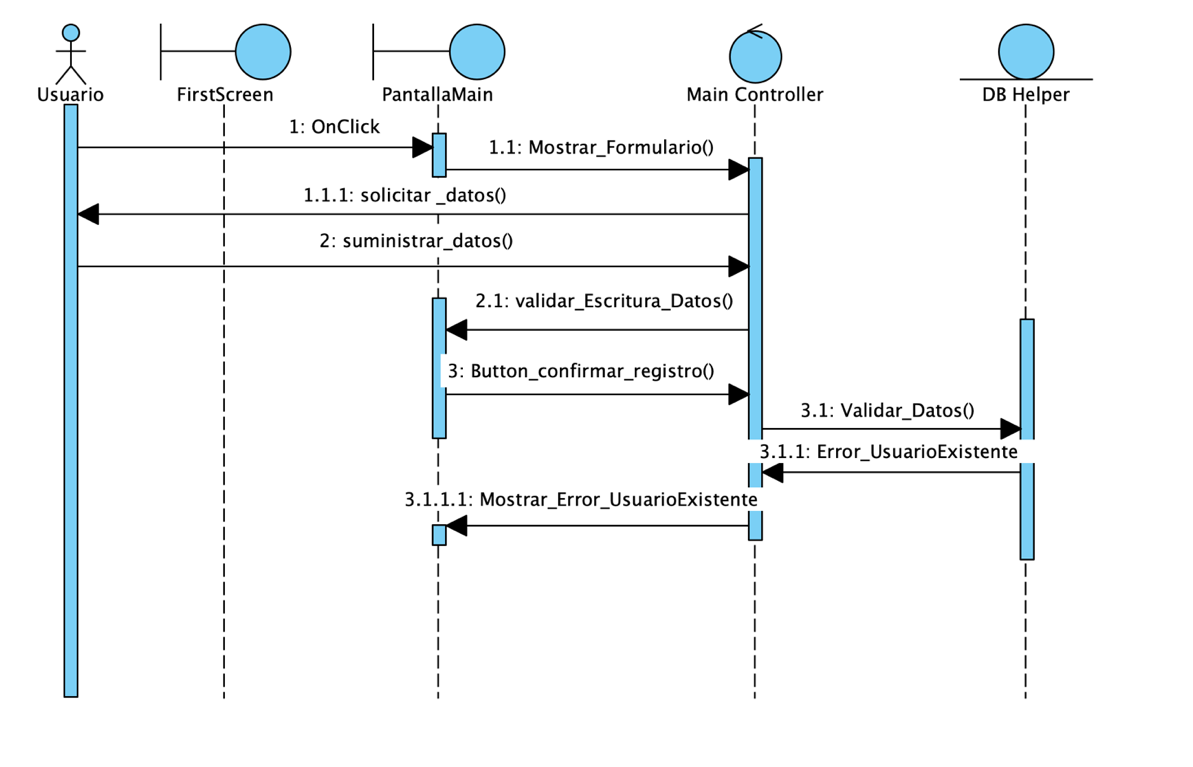
Flujo alterno: Inicio de Sesión



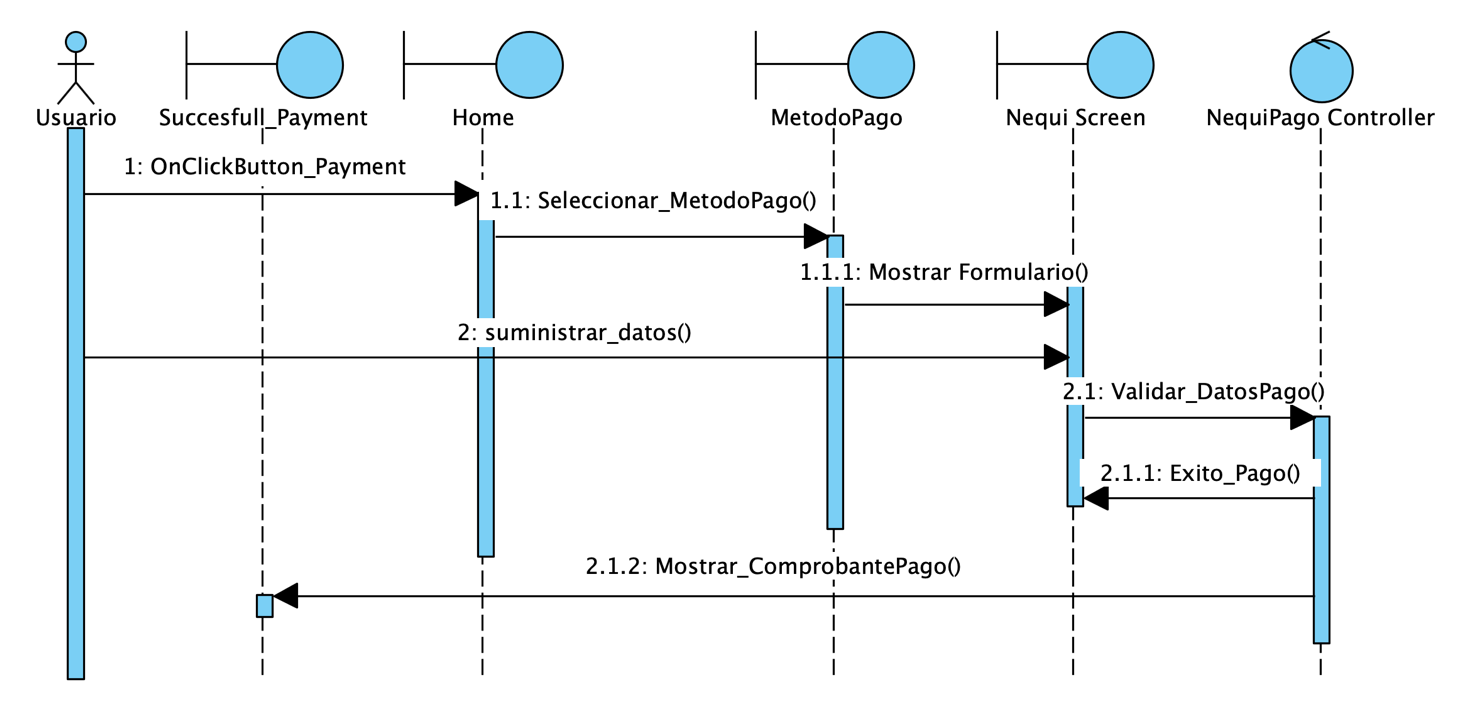
**Registro de usuario: D**iagrama de secuencia en el cual el usuario ingresa a la pantalla principal y selección registrar usuario, en la aplicación cambia de pantalla a “pantalla Main”y procede a pedir los datos del usuario: nombre de usuario, correo y contraseña, la estructura de estos datos es validada por el main controler para revisar que todos los campos estén completos y con las especificaciones necesarias, posterior a esto el controlador de Main manda los datos a validar en la bade de datos “DB Helper” y si el usuario no existe le manda confirmación al controller y este permite la creación del usuario dentro de la base de datos, en caso contrario se le notifica un error al usuario de credenciales invalidas.



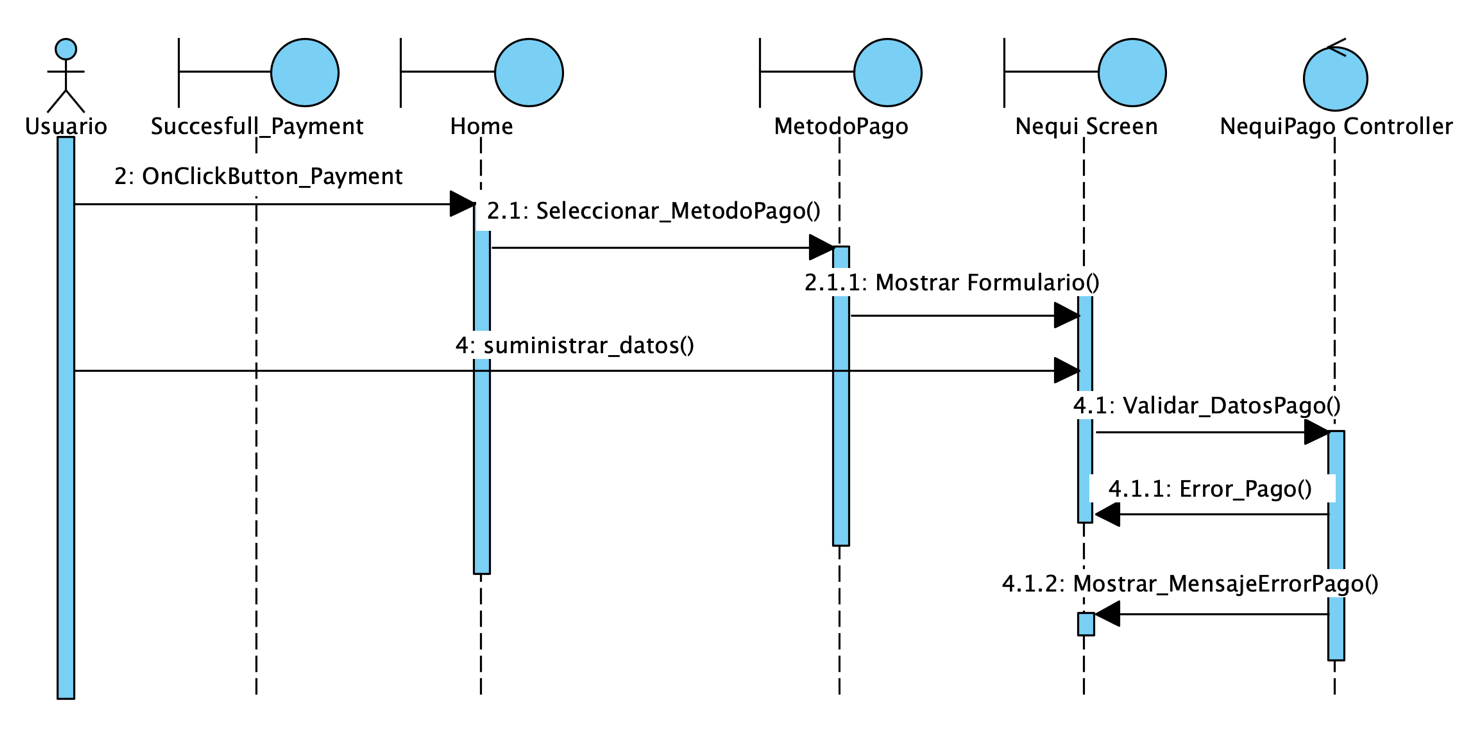
Flujo alterno: Registro de usuario



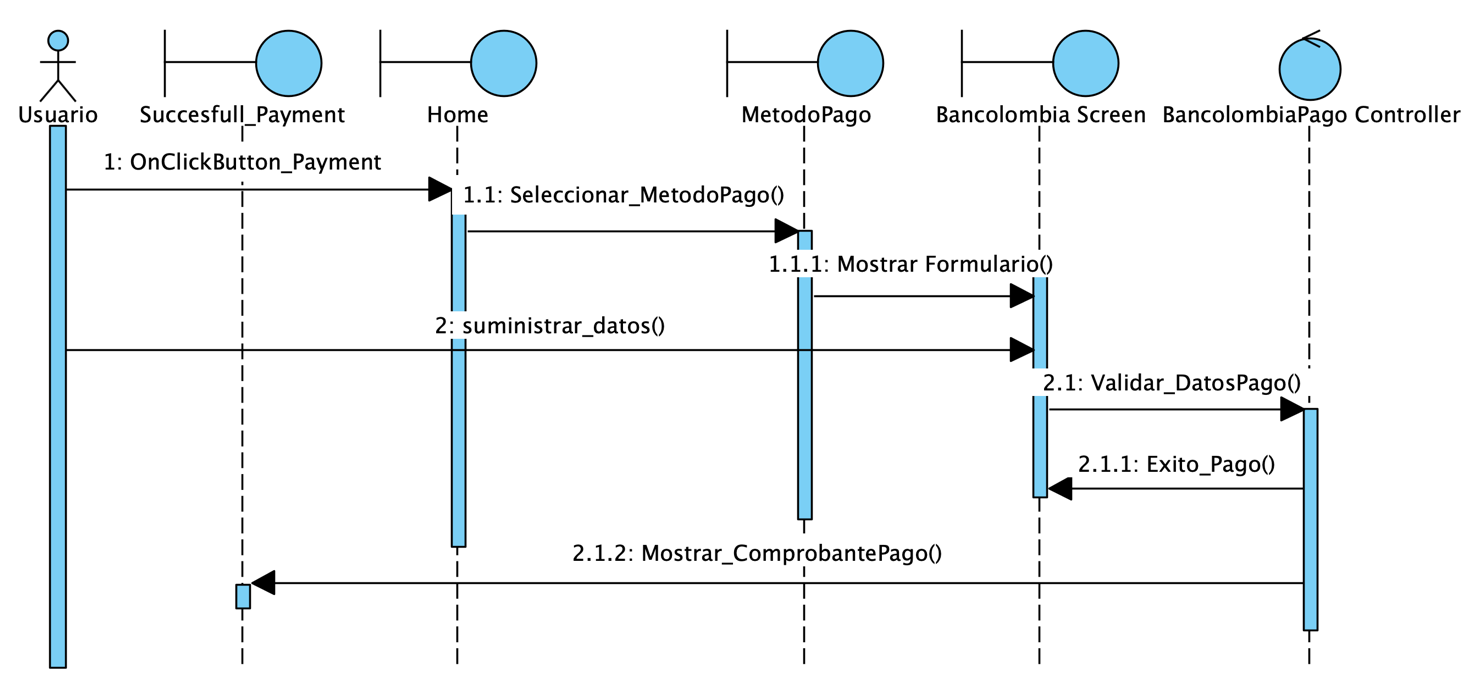
**Pagar Parqueo Nequi:** Diagrama de secuencia en el cual el usuario desea pagar la el valor del estacionamiento con Nequi para esto, se dirige a la pantalla de Home y selecciona Pago, en la aplicación se cambia de pantalla a “Método pago” para que el usuario seleccione el método que desea usar y luego procede a enviarlo a la pantalla de la plataforma Nequi y se procede pedir los datos de pago del usuario: Numero de celular y contraseña, posterior a esto el controlador de Pago valida estos datos y si todo es correcto y la transacción pudo ser realizada le manda confirmación y el usuario es trasladado a la pantalla “Succesfull\_payment” y este le permite ver al usuario su recibo de la transacción, en caso contrario se le notifica un error al usuario de credenciales invalidas.



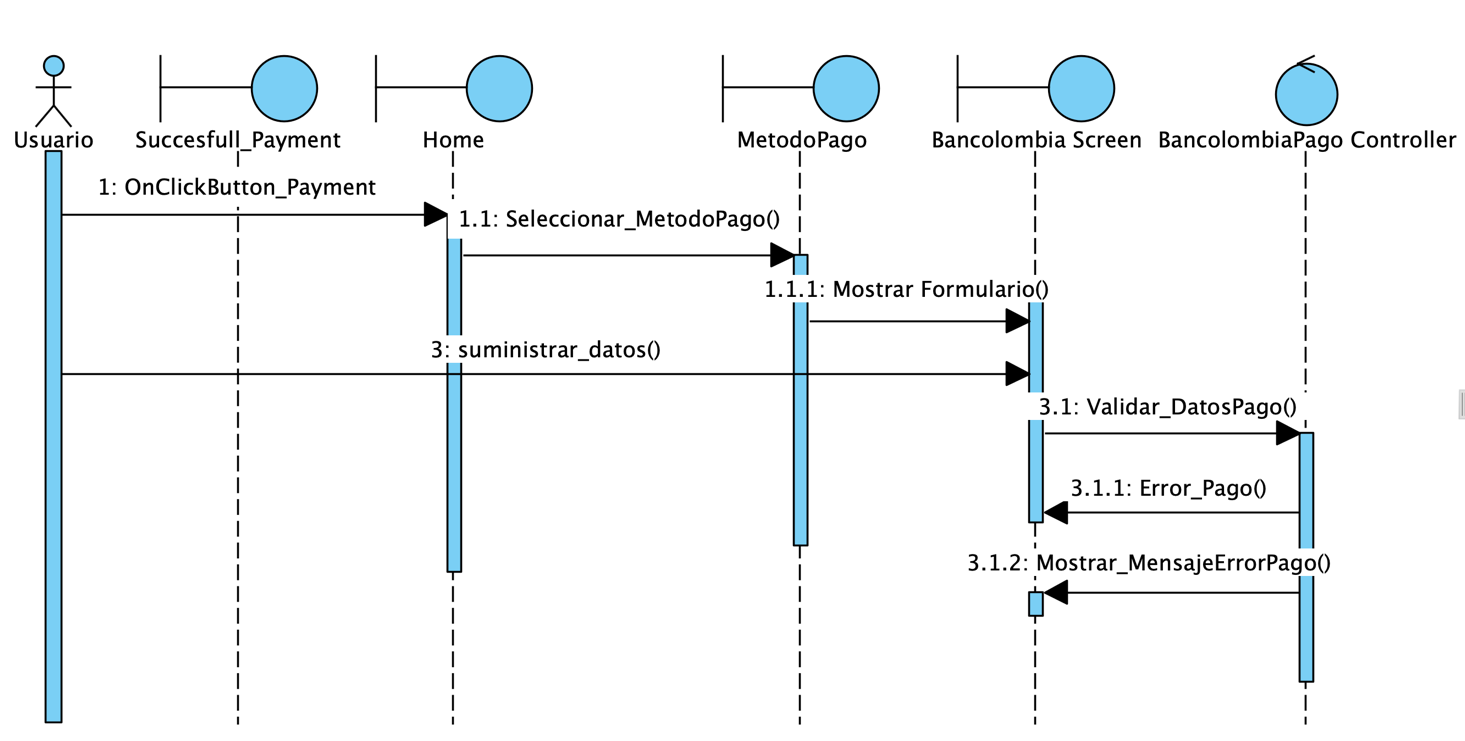
**Flujo alterno: Pago Nequi**

****

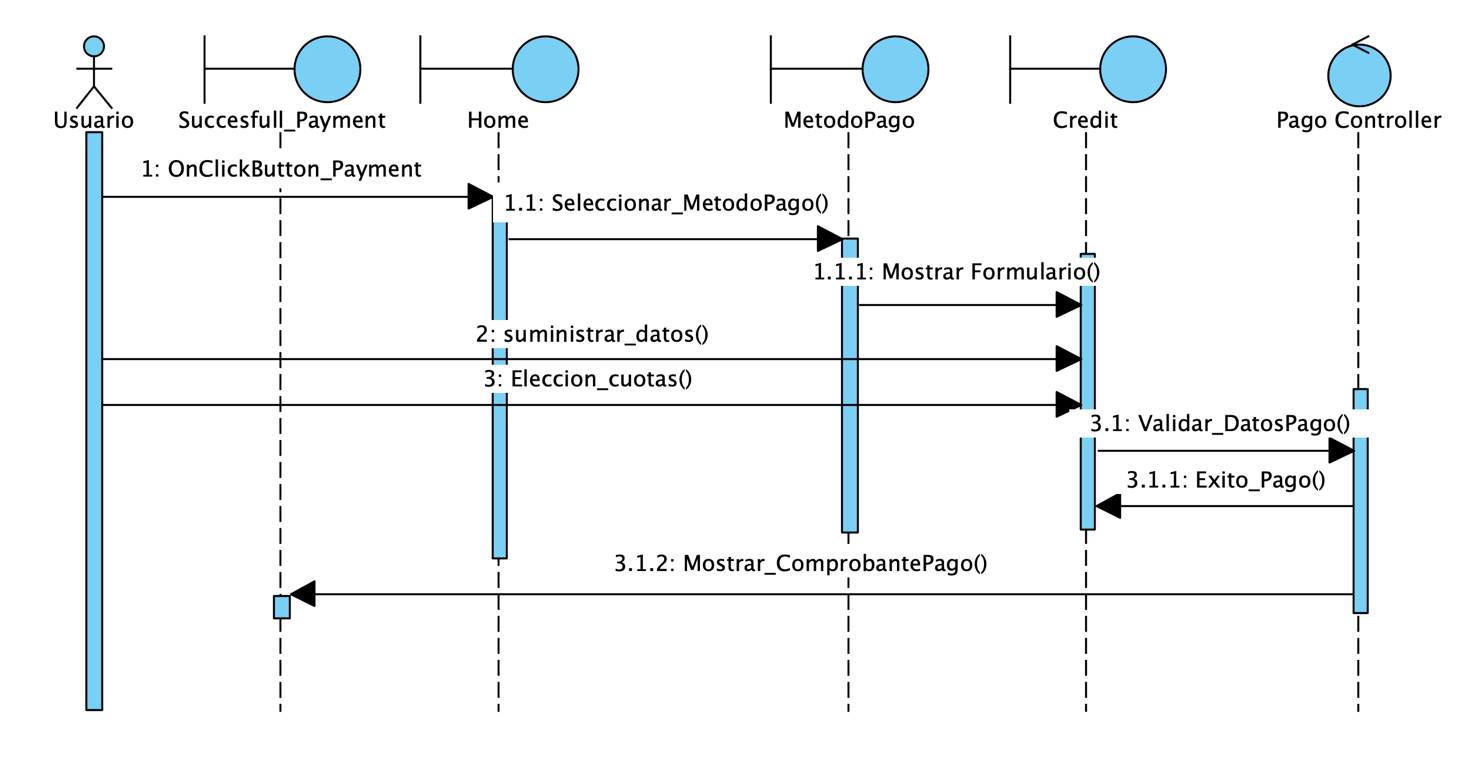
**Pagar Parqueo Bancolombia:** Diagrama de secuencia en el cual el usuario desea pagar la el valor del estacionamiento con Bancolombia para esto, se dirige a la pantalla de Home y selecciona Pago, en la aplicación se cambia de pantalla a “Método pago” para que el usuario seleccione el método que desea usar, en este caso Bancolombia, y luego procede a pedir los datos de pago del usuario, posterior a esto el controlador de Pago valida estos datos y si todo es correcto y la transacción pudo ser realizada le manda confirmación y el usuario es trasladado a la pantalla “Succesfull\_payment” y este le permite ver al usuario su recibo de la transacción, en caso contrario se le notifica un error al usuario de credenciales invalidas.

****

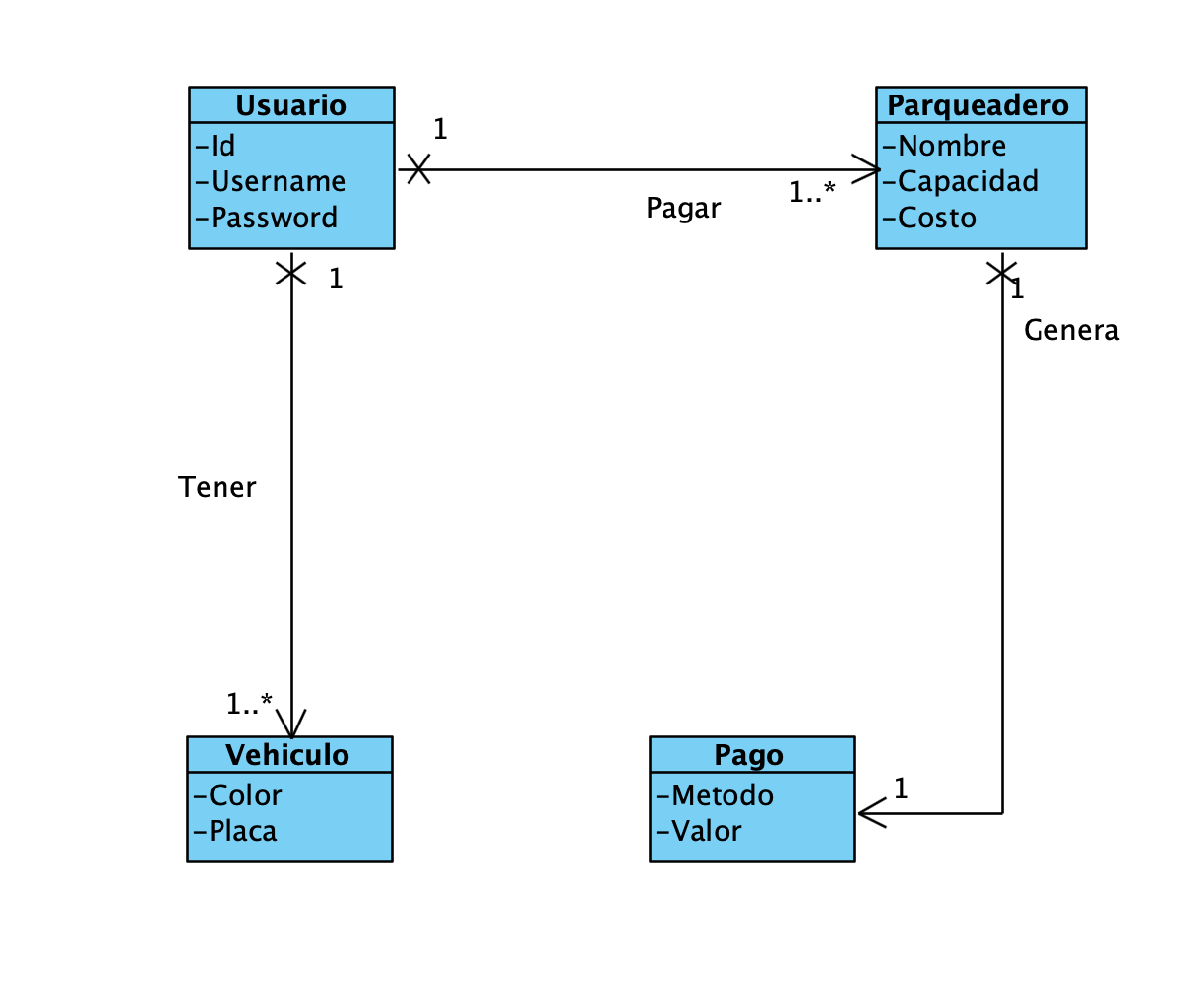
**Flujo alterno: Pago Bancolombia**

****

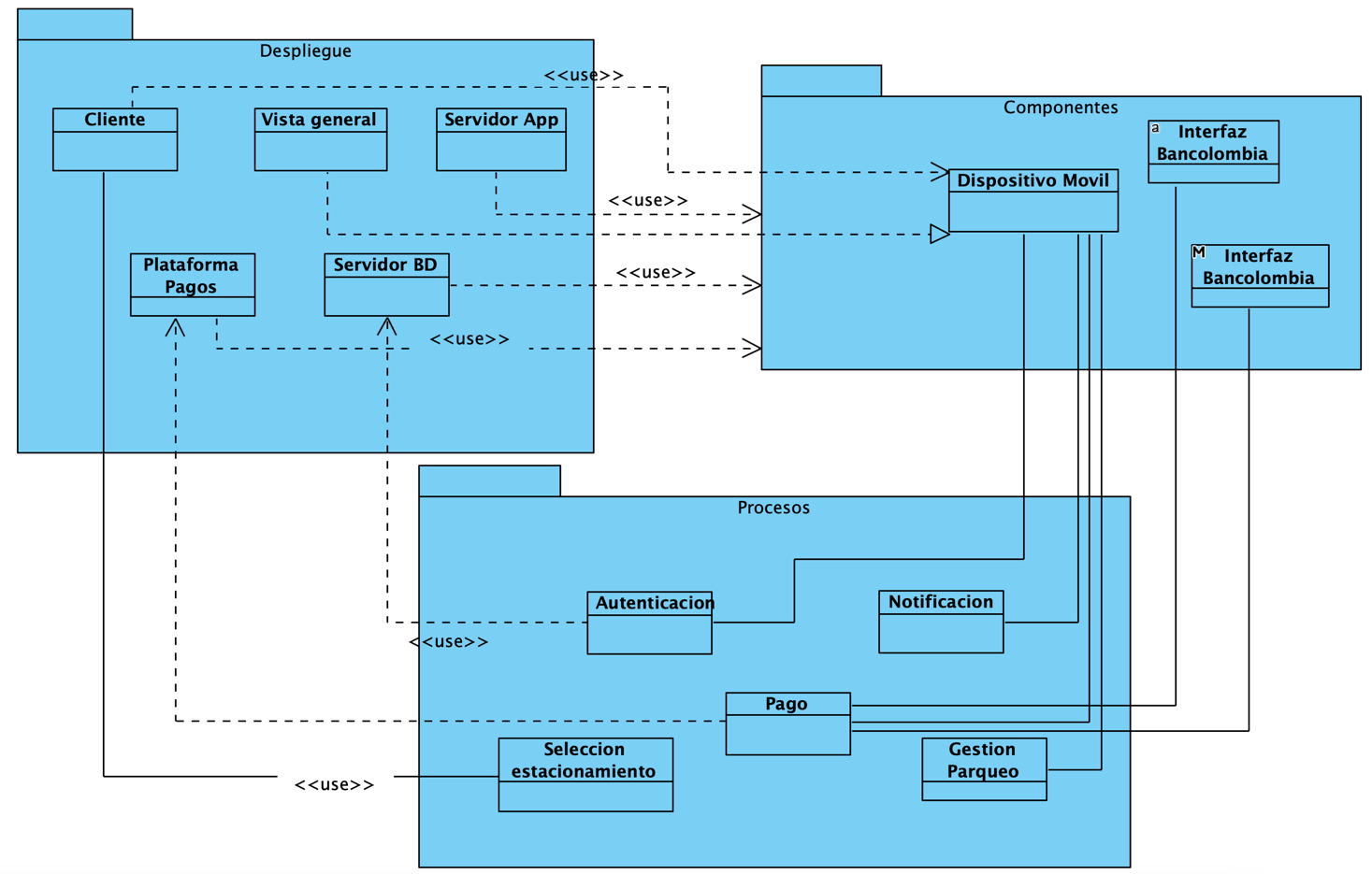
**Pagar Parqueo tarjeta crédito:** Diagrama de secuencia en el cual el usuario desea pagar la el valor del estacionamiento con alguna tarjeta de crédito para esto, se dirige a la pantalla de Home y selecciona Pago, en la aplicación se cambia de pantalla a “Método pago” para que el usuario seleccione el método que desea usar, en este caso tarjeta crédito, y luego procede a pedir los datos de pago del usuario, posterior a esto el controlador de Pago valida estos datos y si todo es correcto y la transacción pudo ser realizada le manda confirmación y el usuario es trasladado a la pantalla “Succesfull\_payment” y este le permite ver al usuario su recibo de la transacción, en caso contrario se le notifica un error al usuario de credenciales invalidas.

****

**Diagrama de Clases:** Diagrama de clases en cual muestra las clases que empleamos dentro de la aplicación para su funcionamiento y caracterización, principalmente las relaciones serian: un usuario puede llegar a tener 1 o muchos vehículos, uno o muchos espacios de estacionamiento ocupados por él y cada uno de estos espacios genera un pago. Finalmente un administrador puede gestionar entre 1 y muchos usuarios y este está encargado de un parqueadero.



**Diagrama MVC:**

****

**Vista Despliegue**

En esta vista de despliegue, podemos identificar los siguientes componentes:

1. Cliente: es el componente que se ejecuta en el dispositivo del usuario (por ejemplo, un teléfono móvil o una tablet) y que le permite interactuar con la aplicación.
2. Servidor de aplicaciones: es el componente que se encarga de procesar las solicitudes del cliente y de coordinar la interacción con los demás componentes. En este caso, el servidor de aplicaciones se encargaría de recibir la solicitud de pago del estacionamiento y de enviarla al componente encargado de procesar los pagos.
3. Servidor de base de datos: es el componente que se encarga de almacenar y gestionar los datos de la aplicación. En este caso, el servidor de base de datos almacenaría la información sobre los pagos realizados por los usuarios.
4. Componente de procesamiento de pagos: es el componente encargado de procesar las transacciones de pago. Este componente se encargaría de comunicarse con los sistemas de pago externos (por ejemplo, con una entidad financiera) para autorizar y registrar el pago del estacionamiento.
5. Vista de despliegue: es la vista que nos muestra cómo se distribuyen los diferentes componentes en el entorno de ejecución. En esta vista podemos ver cómo se conectan los diferentes componentes y cómo se comunican entre sí.

**Vista Componentes**

La vista de componentes nos muestra los diferentes componentes que se ejecutan en el hardware, como por ejemplo:

1. Dispositivo móvil del usuario: es el componente que se ejecuta en el dispositivo del usuario y que le permite interactuar con la aplicación. Este componente puede incluir una aplicación móvil o una aplicación web.
2. Servidor de aplicaciones: es el componente que se encarga de procesar las solicitudes del usuario y de coordinar la interacción con los demás componentes. En este caso, el servidor de aplicaciones se encargaría de recibir la solicitud de pago del estacionamiento y de enviarla al componente encargado de procesar los pagos.
3. Servidor de base de datos: es el componente que se encarga de almacenar y gestionar los datos de la aplicación. En este caso, el servidor de base de datos almacenaría la información sobre los pagos realizados por los usuarios.
4. Componente de procesamiento de pagos: es el componente encargado de procesar las transacciones de pago. Este componente se encargaría de comunicarse con los sistemas de pago externos (por ejemplo, con una entidad financiera) para autorizar y registrar el pago del estacionamiento.

**Vista Procesos**

La vista de procesos nos muestra los diferentes procesos que se ejecutan en la aplicación y cómo se relacionan entre sí para brindar al usuario una experiencia fluida. Algunos de los procesos que podemos identificar en esta vista son los siguientes:

1. Proceso de autenticación: es el proceso que se encarga de autenticar al usuario y de validar su identidad. Este proceso es fundamental para garantizar la seguridad de la aplicación y de los datos de los usuarios.
2. Proceso de selección del estacionamiento: es el proceso que se encarga de mostrar al usuario los diferentes estacionamientos disponibles y de permitirle seleccionar el que desea utilizar.
3. Proceso de pago: es el proceso que se encarga de procesar el pago del estacionamiento. Este proceso se comunica con el componente de procesamiento de pagos para autorizar y registrar el pago del usuario.
4. Proceso de notificación: es el proceso que se encarga de enviar notificaciones al usuario sobre el estado del pago y de la duración del estacionamiento.
5. Proceso de gestión de estacionamientos: es el proceso que se encarga de gestionar la disponibilidad de los estacionamientos y de actualizar su estado en tiempo real.