

# Memoria Proyecto Final



**Mario Clavero Artal 875371**  
**Andrea Ferradas Agudo 873635**  
**Grupo: M35**  
**Curso 2024/2025**  
**Ingeniería del Software**

# Índice

<b>1. Resumen</b>	<b>3</b>
<b>2. Introducción y objetivos</b>	<b>3</b>
<b>3. Catálogo De Requisitos</b>	<b>4</b>
<b>4. Casos De Uso</b>	<b>5</b>
Diagrama De Casos De Uso	5
Matriz Relacional Casos De Uso con Requisitos Funcionales	6
Descripción De Cada Caso De Uso	6
<b>5. Análisis</b>	<b>13</b>
Modelado Estático: Diagrama De Clases	14
Modelado Dinámico: Diagramas De Secuencias	15
<b>6. Prototipado De La Interfaz De Usuario</b>	<b>21</b>
Prototipado De Pantallas	21
Mapa De Navegación Del Prototipo	25
Matriz Relacional Casos De Uso con Pantallas	26
<b>7. Diseño del Sistema y Diseño de Objetos</b>	<b>27</b>
Modelo lógico de la base de datos relacional	27
Diagrama de paquetes	28
Diagrama de componentes	29
Diagrama de despliegue	29
Diagrama de clases	30
Diagramas de secuencia a nivel de diseño	31
<b>8. Implementación</b>	<b>43</b>
Prototipado De Pantallas Finales	43
Mapa De Navegación Final	46
<b>9. Pruebas</b>	<b>47</b>

## **1. Resumen**

Esta práctica consiste en la realización de una aplicación que lleva la gestión de las reservas de las parcelas de un camping. Cada reserva tiene diversas parcelas asociadas y distinto número de ocupantes según la reserva.

## **2. Introducción y objetivos**

En la primera práctica se va a realizar la recogida de los diversos requisitos tanto funcionales como no funcionales de la aplicación, además del diccionario de datos. Despues, se elaborará un diagrama de casos de uso que recogerá todo lo que el administrador podrá hacer en la aplicación, asimismo cada caso de uso constará de una descripción.

En la parte de análisis se realizará un diagrama de clases para la parte del modelado estático, mientras que en la parte del modelado dinámico se harán diagramas de secuencias para todos los casos de uso comentados en el apartado anterior.

### 3. Catálogo De Requisitos

Para la realización de la aplicación se necesita definir un catálogo con los requisitos tanto funcionales como no funcionales que debe poseer. Además, se describen las restricciones y el diccionario de datos, en donde se encuentran definidas las dos entidades principales.

Requisitos Funcionales	
Código	Descripción
<b>RF-1</b>	El sistema tiene que poder crear parcelas
<b>RF-2</b>	El sistema debe permitir a los usuarios poder consultar una lista con las parcelas creadas previamente. Pudiendo ser ordenada por identificador, número máximo de ocupantes o precio.
<b>RF-3</b>	El sistema debe poder modificar las parcelas que se habían creado previamente.
<b>RF-4</b>	El sistema debe permitir que la eliminación de parcelas previamente creadas
<b>RF-5</b>	El sistema debe permitir la creación de una reserva
<b>RF-6</b>	El sistema debe permitir la consulta de un listado con todas las reservas previamente creadas. Pudiendo ordenarlas por nombre de cliente, número de móvil o fecha de entrada.
<b>RF-7</b>	El sistema debe permitir la modificación de las reservas ya creadas
<b>RF-8</b>	El sistema permite eliminar reservas creadas previamente
<b>RF-9</b>	El sistema permite enviar al móvil del cliente toda la información sobre la reserva cuando el propietario lo considere oportuno
Requisitos No Funcionales	
Código	Descripción
<b>RNF-1</b>	El sistema debe seguir y cumplir las Leyes de Protección de Datos.
<b>RNF-2</b>	El sistema debe funcionar en dispositivos móviles con sistemas operativos Android 14.0.
<b>RNF-3</b>	El sistema debe permitir crear hasta 100 parcelas.
<b>RNF-4</b>	El sistema debe permitir almacenar hasta 10000 reservas en la aplicación.
Diccionario de Datos	

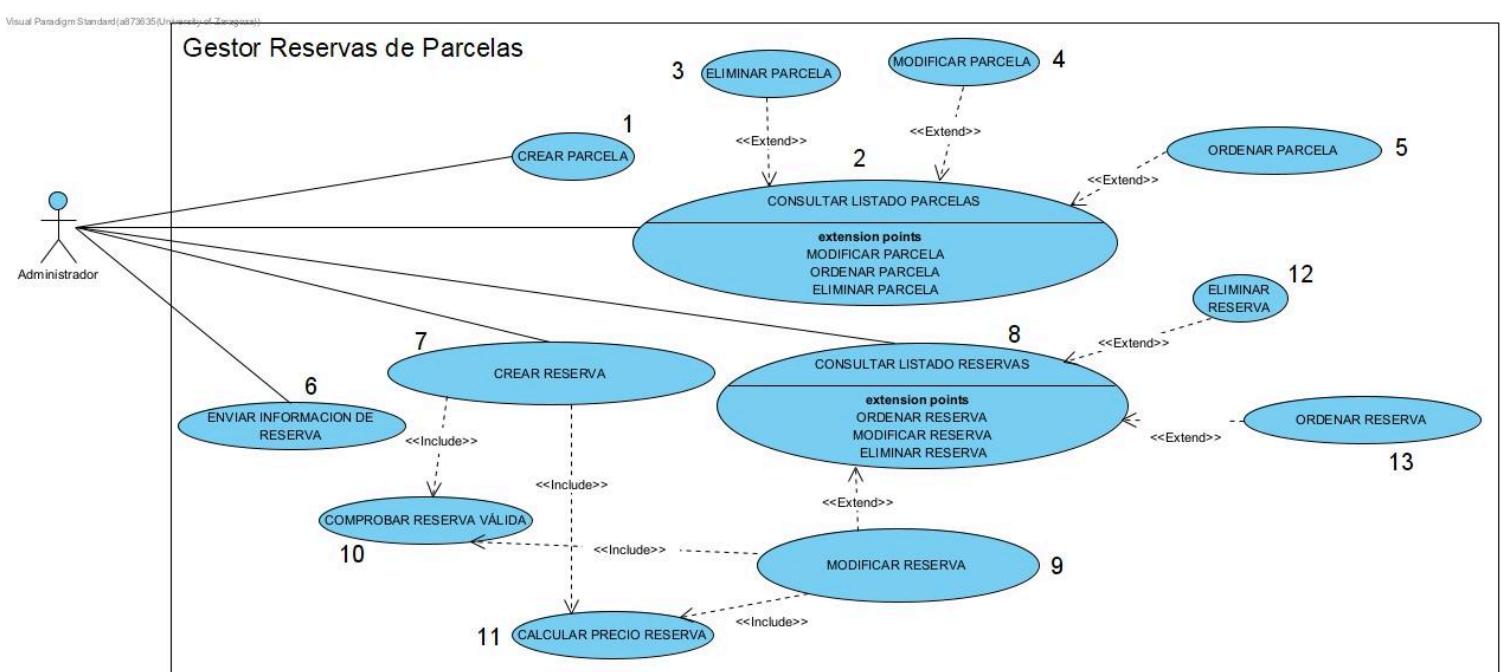
<b>Parcela</b>	Se identifica con el nombre de un monte o valle cercano. Costará de una descripción en texto libre de las características de la parcela, el número máximo de ocupantes y el precio por persona
<b>Reserva</b>	Consta de un nombre de cliente, número móvil del cliente, fecha de entrada y salida, y las parcelas reservadas con el número de ocupantes en cada una. El precio total de la reserva se mantendrá aunque posteriormente se modifique el precio por persona de la parcela.

## 4. Casos De Uso

En este apartado se describe cómo interactúan los usuarios con el sistema a través de los diferentes casos de uso. Se incluye un diagrama que ilustra de forma visual las interacciones principales, una matriz relacional que asocia cada caso de uso con los requisitos funcionales, y una descripción detallada de cada caso de uso para comprender su alcance y funcionalidad.

### Diagrama De Casos De Uso

El diagrama de casos de uso proporciona una representación gráfica de las interacciones entre los actores (usuarios del sistema) y las funcionalidades principales de la aplicación. Este diagrama permite visualizar de manera sencilla qué acciones pueden realizar los usuarios y cómo se relacionan con el sistema, facilitando la comprensión de los procesos que aborda la aplicación.



## Matriz Relacional Casos De Uso con Requisitos Funcionales

La matriz relacional establece una correspondencia directa entre los casos de uso y los requisitos funcionales definidos en el apartado anterior, con ella se pueden observar más fácilmente qué requisitos se cubren con cada caso de uso.

Casos de Uso \ Requisitos	1	2	3	4	5	6	7	8	9
1	X								
2		X							
3				X					
4			X						
5		X							
6									X
7				X					
8					X				
9						X			
10									
11									
12								X	
13					X				

## Descripción De Cada Caso De Uso

En este subapartado se detalla cada caso de uso individualmente, definiendo aspectos clave como los actores involucrados y el flujo de eventos. Estas descripciones ofrecen una mejor visión de cómo se espera que el sistema funcione.

### → Crear parcela

- ◆ Caso de uso: Crear parcela
- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de crear parcela.
  - El administrador tiene que llenar los campos de nombre, descripción, número máximo de ocupantes y precio por persona.
  - Confirma la creación de la parcela tras llenar todos los campos.

- El sistema le muestra un mensaje de confirmación tras crear la parcela.
- Termina cuando el administrador vuelve a la pantalla de inicio o cierra la aplicación.
- ◆ Flujo de eventos alternativo
  - Comienza cuando el nombre introducido para crear la parcela ya existe.
  - El sistema muestra un mensaje de error ya que no puede haber parcelas con nombres repetidos.
  - El administrador rellena el campo nombre con otro no existente y pulsa en confirmar la creación de la parcela.
  - Termina cuando el sistema muestra el mensaje de confirmación.
- ◆ Flujo de eventos alternativo
  - Comienza cuando el administrador no ha llenado todos los campos necesarios.
  - El sistema muestra un mensaje de error ya que se necesitan todos los campos.
  - El administrador introduce todos los datos y confirma la creación de la parcela.
  - Termina cuando el sistema muestra el mensaje de confirmación.

## → Consultar parcelas

- ◆ Caso de uso: Consultar listado parcelas
- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de consultar listado parcelas.
  - El sistema muestra el listado de las parcelas.
  - El administrador consulta el listado, pudiendo realizar las acciones:
    - *extend*: eliminar parcela
    - *extend*: modificar parcela
    - *extend*: ordenar parcela (por identificador, por número máximo de ocupantes o por precio)
  - Termina cuando el administrador vuelve a la pantalla de inicio o cierra la aplicación.
- ◆ Flujo de eventos alternativo
  - Comienza cuando el administrador termina de eliminar o modificar una parcela.
  - El sistema muestra el listado de las parcelas después de haber eliminado o modificado una parcela.

## → Eliminar parcela

- ◆ Caso de uso: Eliminar parcela
- ◆ Actor: Administrador

- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de consultar listado parcelas.
  - El sistema muestra el listado de las parcelas, con opciones de eliminar parcela y modificar parcela.
  - El administrador pulsa la opción de eliminar parcela.
  - El sistema muestra un mensaje “Esta acción es irreversible, ¿está seguro que quiere eliminar la parcela?” con un botón para confirmar y otro para cancelar.
  - El administrador pulsa la opción de confirmar.
  - El caso de uso termina cuando el sistema muestra un mensaje “Parcela eliminada correctamente” y vuelve a mostrar el listado de las parcelas.
- ◆ Flujo de eventos alternativo
  - Comienza cuando el administrador pulsa el botón para cancelar la eliminación de la parcela seleccionada.
  - El caso de uso finaliza con el sistema mostrando el listado de las parcelas sin modificaciones.

#### → Modificar parcela

- ◆ Caso de uso: Modificar parcela
- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de consultar listado parcelas.
  - El sistema muestra el listado de las parcelas, con opciones de eliminar parcela y modificar parcela.
  - El administrador pulsa la opción de modificar parcela.
  - El sistema muestra los campos de nombre, descripción, número máximo de ocupantes y precio por persona, con la información introducida anteriormente.
  - El administrador modifica los campos que deseé y pulsa el botón de confirmar cambios.
  - El sistema muestra un mensaje “Esta acción es irreversible, ¿está seguro que quiere modificar la parcela?” con un botón para confirmar y otro para cancelar.
  - El administrador pulsa la opción de confirmar.
  - El caso de uso termina cuando el sistema muestra un mensaje “Parcela modificada correctamente” y vuelve a mostrar el listado de las parcelas.
- ◆ Flujo de eventos alternativo
  - Comienza cuando el administrador pulsa el botón para cancelar la modificación de la parcela seleccionada.
  - El caso de uso finaliza con el sistema mostrando el listado de las parcelas sin modificaciones.

## → Ordenar parcela

- ◆ Caso de uso: Ordenar parcela
- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de consultar listado parcelas.
  - El sistema muestra el listado de las parcelas, con una opción de ordenarlas según un criterio a elección (por identificador, por número máximo de ocupantes o por precio).
  - El administrador pulsa una de las opciones de ordenación.
  - Termina cuando el sistema muestra el listado de las parcelas ordenado según el criterio elegido por el administrador.

## → Crear reserva

- ◆ Caso de uso: Crear reserva
- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de crear reserva.
  - El administrador rellena los campos de nombre del cliente, teléfono del cliente, fecha de entrada, fecha de salida, precio total y los identificadores de las parcelas reservadas con el número de ocupantes en cada una.
  - Confirma la creación de la reserva tras llenar todos los campos.
    - *include*: comprobar reserva válida
    - *include*: calcular precio reserva
  - El caso de uso termina cuando el sistema muestra un mensaje de confirmación tras crear la reserva.
- ◆ Flujo de eventos alternativo
  - Comienza cuando el administrador confirma la creación de la reserva.
    - *include*: comprobar reserva válida
    - *include*: calcular precio reserva
  - El sistema muestra un mensaje de error indicando los motivos del error al crear la reserva: se supera el máximo de ocupantes de una/varias parcela/s, la fecha de entrada es posterior a la fecha de salida, las parcelas introducidas ya están ocupadas en esas fechas.
  - El administrador modifica los campos incorrectos y confirma la creación de la reserva.
    - *include*: comprobar reserva válida
    - *include*: calcular precio reserva
  - El caso de uso termina cuando los datos introducidos son correctos, por lo que se vuelve al flujo de eventos principal, o cuando son incorrectos, por ello comienza uno de los flujos de eventos alternativo.
- ◆ Flujo de eventos alternativo

- Comienza cuando el administrador confirma la creación de la reserva.
  - *include*: comprobar reserva válida
  - *include*: calcular precio reserva
- El sistema muestra un mensaje de error ya que no se han llenado todos los campos.
- El administrador introduce todos los datos y confirma la creación de la reserva.
  - *include*: comprobar reserva válida
  - *include*: calcular precio reserva
- Termina cuando el sistema muestra el mensaje de confirmación.

#### → Enviar información reserva

- ◆ Caso de uso: Enviar información de reserva
- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de enviar información de reserva.
  - El sistema envía toda la información de la reserva al número de teléfono asociado del cliente.
  - El caso de uso termina cuando el sistema muestra un mensaje “Información enviada al cliente”.

#### → Consultar listado reservas

- ◆ Caso de uso: Consultar listado reservas
- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de consultar listado reservas.
  - El sistema muestra el listado de las reservas.
  - El administrador consulta el listado, pudiendo realizar las acciones:
    - *extend*: eliminar reserva
    - *extend*: modificar reserva
    - *extend*: ordenar reserva (por nombre del cliente, por número de teléfono del cliente o por fecha de entrada)
  - Termina cuando el administrador vuelve a la pantalla de inicio o cierra la aplicación.
- ◆ Flujo de eventos alternativo
  - Comienza cuando el administrador termina de eliminar o modificar una reserva.
  - El sistema muestra el listado de las reservas después de haber eliminado o modificado una reserva.

#### → Eliminar reserva

- ◆ Caso de uso: Eliminar reserva

- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de consultar listado reservas.
  - El sistema muestra el listado de las reservas, con opciones de eliminar parcela y modificar reserva.
  - El administrador pulsa la opción de eliminar reserva.
  - El sistema muestra un mensaje “Esta acción es irreversible, ¿está seguro que quiere eliminar la reserva?” con un botón para confirmar y otro para cancelar.
  - El administrador pulsa la opción de confirmar.
  - El caso de uso termina cuando el sistema muestra un mensaje “Reserva eliminada correctamente” y vuelve a mostrar el listado de las reservas.
- ◆ Flujo de eventos alternativo
  - Comienza cuando el administrador pulsa el botón para cancelar la eliminación de la reserva seleccionada.
  - El caso de uso finaliza con el sistema mostrando el listado de las reservas sin modificaciones.

#### → Ordenar reserva

- ◆ Caso de uso: Ordenar reserva
- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de consultar listado reservas.
  - El sistema muestra el listado de las reservas, con una opción de ordenarlas según un criterio a elección (por nombre del cliente, por número de teléfono del cliente o por fecha de entrada).
  - El administrador pulsa una de las opciones de ordenación.
  - Termina cuando el sistema muestra el listado de las reservas ordenado según el criterio elegido por el administrador.

#### → Modificar reserva

- ◆ Caso de uso: Modificar reserva
- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de consultar listado reservas.
  - El sistema muestra el listado de las reservas, con opciones de eliminar parcela y modificar reserva.
  - El administrador pulsa la opción de modificar reserva.
  - El sistema muestra los campos de nombre del cliente, teléfono del cliente, fecha de entrada, fecha de salida, precio total y los

identificadores de las parcelas reservadas con el número de ocupantes en cada una; con la información introducida anteriormente.

- El administrador modifica los campos que deseé y pulsa el botón de confirmar cambios.
- El sistema muestra un mensaje “Esta acción es irreversible, ¿está seguro que quiere modificar la reserva?” con un botón para confirmar y otro para cancelar.
- El administrador pulsa la opción de confirmar.
  - *include*: comprobar reserva válida
  - *include*: calcular precio reserva
- El caso de uso termina cuando el sistema muestra un mensaje “Reserva modificada correctamente” y vuelve a mostrar el listado de las reservas.

◆ Flujo de eventos alternativo

- Comienza cuando el administrador pulsa el botón para cancelar la modificación de la reserva seleccionada.
- El caso de uso finaliza con el sistema mostrando el listado de las reservas sin modificaciones.

◆ Flujo de eventos alternativo

- Comienza cuando el administrador confirma la modificación de la reserva.
  - *include*: comprobar reserva válida
  - *include*: calcular precio reserva
- El sistema muestra un mensaje de error indicando los motivos del error al modificar la reserva: se supera el máximo de ocupantes de una/varias parcela/s, la fecha de entrada es posterior a la fecha de salida, las parcelas introducidas ya están ocupadas en esas fechas.
- El administrador modifica los campos incorrectos y confirma la creación de la reserva.
  - *include*: comprobar reserva válida
  - *include*: calcular precio reserva
- El caso de uso termina cuando los datos introducidos son correctos, por lo que se vuelve al flujo de eventos principal, o cuando son incorrectos, por ello comienza uno de los flujos de eventos alternativo.

◆ Flujo de eventos alternativo

- Comienza cuando el administrador confirma la creación de la reserva.
  - *include*: comprobar reserva válida
  - *include*: calcular precio reserva
- El sistema muestra un mensaje de error ya que no se han llenado todos los campos.
- El administrador introduce todos los datos y confirma la creación de la reserva.
  - *include*: comprobar reserva válida
  - *include*: calcular precio reserva

- Termina cuando el sistema muestra el mensaje de confirmación.

#### → Comprobar reserva válida

- ◆ Caso de uso: Comprobar reserva válida
- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de crear reserva o cuando selecciona modificar reserva desde el listado de reservas.
  - El sistema comprueba si son correctos los datos introducidos de la reserva: no están ocupadas las parcelas introducidas en las fechas seleccionadas, la fecha de entrada es anterior a la fecha de salida y el número de ocupantes de cada parcela no supera su máximo.
  - El sistema verifica que sí son correctos.
  - El caso de uso termina cuando el sistema muestra un mensaje “Reserva correcta” de confirmación.
- ◆ Flujo de eventos alternativo
  - Comienza cuando el sistema comprueba los datos introducidos.
  - El caso de uso termina cuando el sistema muestra un mensaje de error ya que los datos incumplen alguna de las condiciones necesarias y devuelve con error al administrador al caso de uso del que provenía (crear reserva o modificar reserva).

#### → Calcular precio reserva

- ◆ Caso de uso: Calcular precio reserva
- ◆ Actor: Administrador
- ◆ Flujo de eventos principal
  - Comienza cuando el administrador selecciona la opción de crear reserva o cuando selecciona modificar reserva desde el listado de reservas.
  - El sistema calcula de forma automática el precio de la reserva según el número de ocupantes de una parcela y su precio por persona.
  - El caso de uso termina cuando el sistema devuelve el valor del precio calculado.

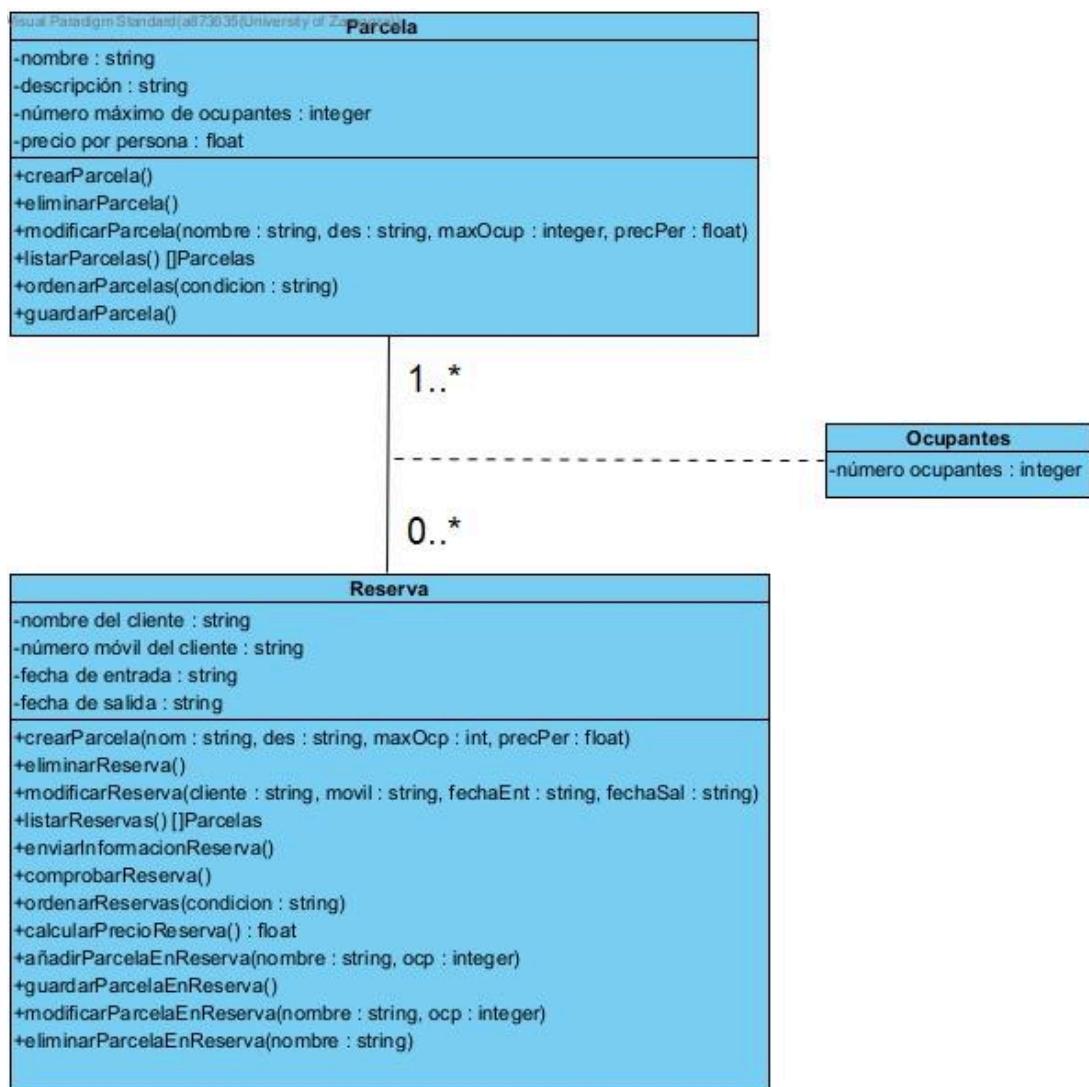
## 5. Análisis

En este apartado se presenta un análisis detallado de la estructura y el comportamiento del sistema, utilizando herramientas de modelado estático y dinámico para comprender mejor su funcionamiento y diseño. Estos modelos permiten definir cómo se organizan los datos y cómo interactúan los diferentes

elementos del sistema en tiempo de ejecución, facilitando así su implementación y validación.

## Modelado Estático: Diagrama De Clases

El diagrama de clases representa la estructura estática del sistema, mostrando las principales clases, sus atributos, métodos y las relaciones entre ellas. Con este modelo podemos tener una visión más clara de cómo se ordenan las parcelas y las reservas, sus atributos y sus funciones.



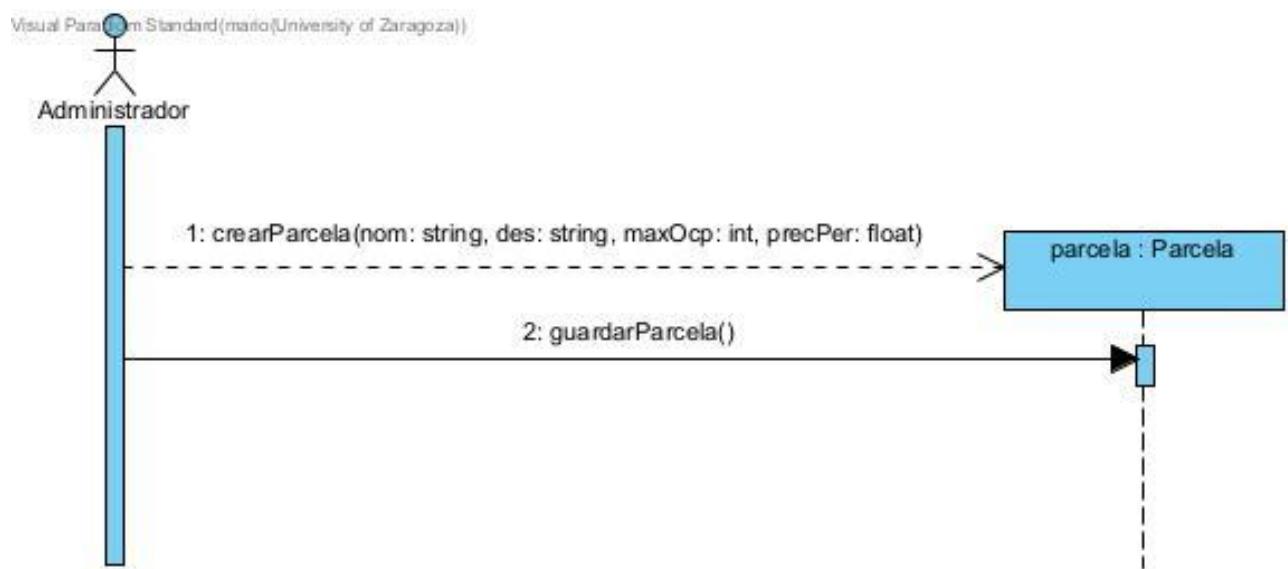
Para el diagrama se han utilizado tres clases: Parcela, Reserva y Ocupantes. De esta forma se relaciona una reserva con varias parcelas y se asocia el número de ocupantes por parcela; y una parcela tiene múltiples asociaciones con reservas y los ocupantes asociados.

- Parcela: Almacena el nombre, una descripción, el número máximo de ocupantes y el precio por persona.
- Reserva: Guarda del cliente su nombre y su número de teléfono, la fecha de entrada y de salida. Además se almacena el precio total, calculado a partir de una función auxiliar.
- Ocupantes: Almacena los ocupantes que se alojan en una parcela concreta en una reserva.

## Modelado Dinámico: Diagramas De Secuencias

Para todos los casos de uso descritos anteriormente se ha realizado un diagrama de secuencia. En cada uno de ellos se describe el flujo de eventos que tendrá lugar en cada caso de uso.

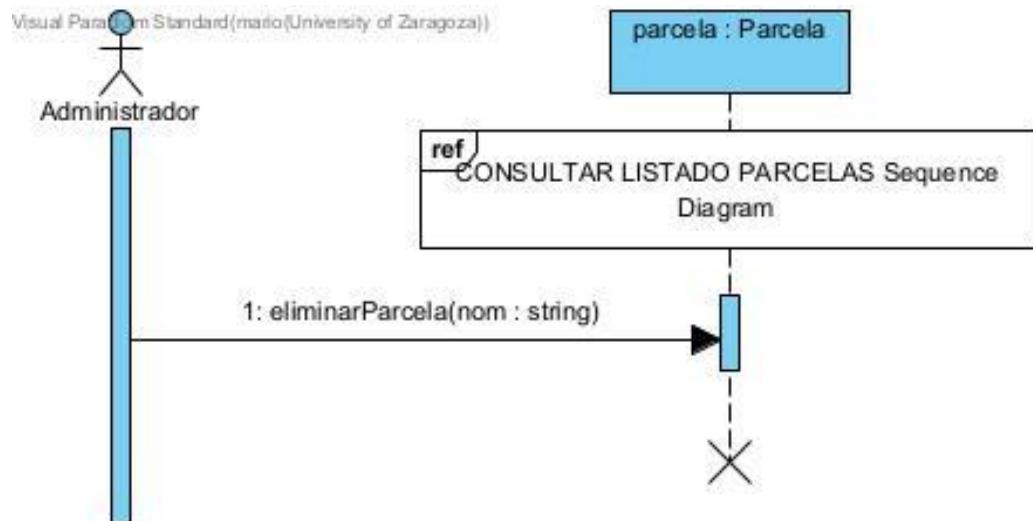
### → Crear parcela



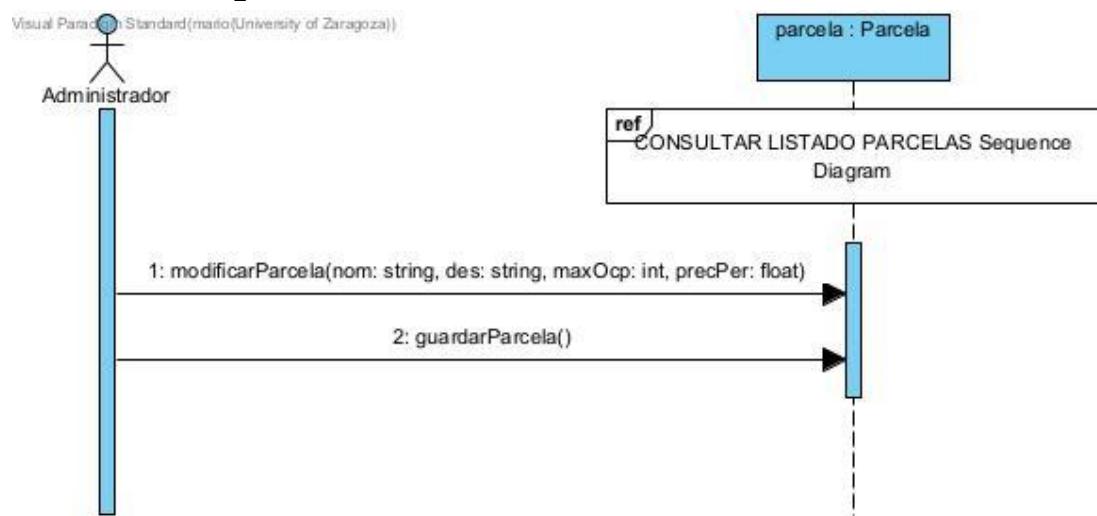
### → Consultar parcelas



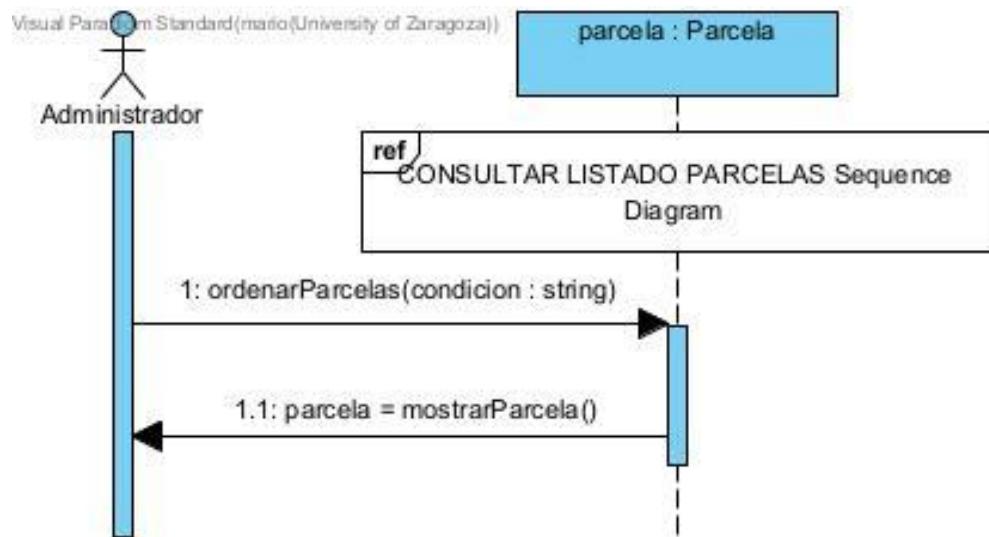
### → Eliminar parcela



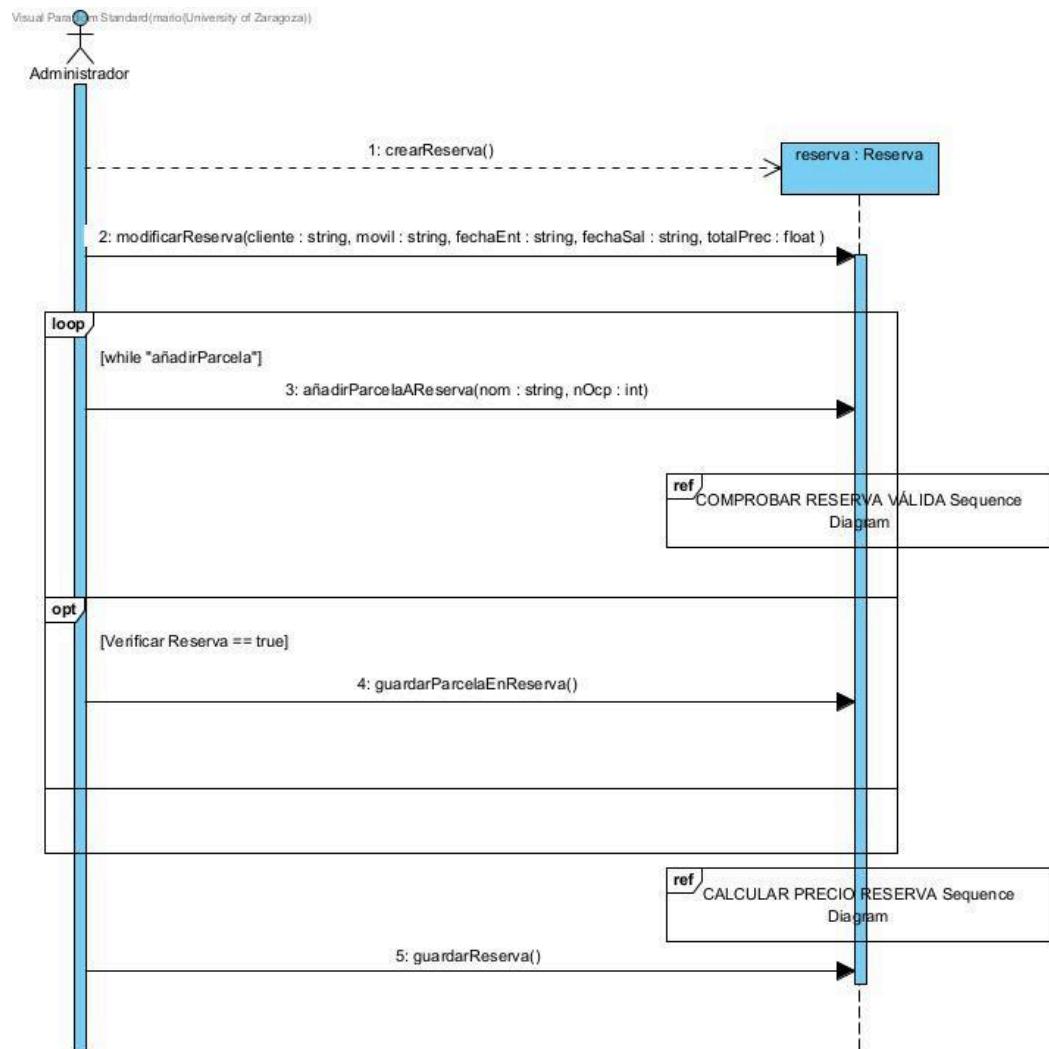
### → Modificar parcela



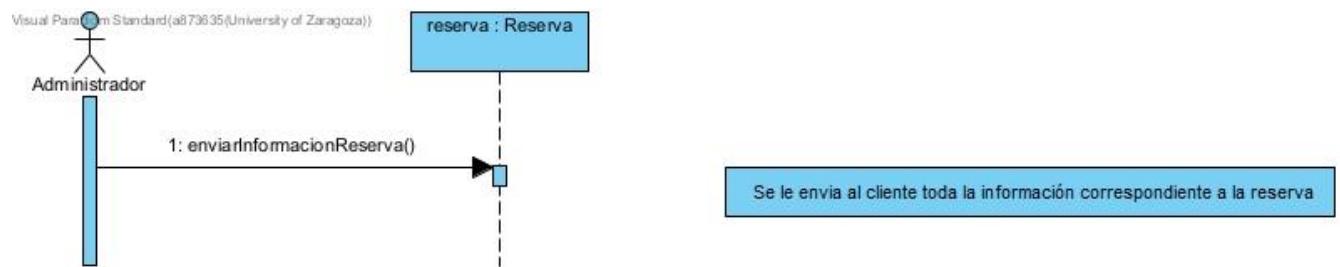
## → Ordenar parcela



## → Crear reserva



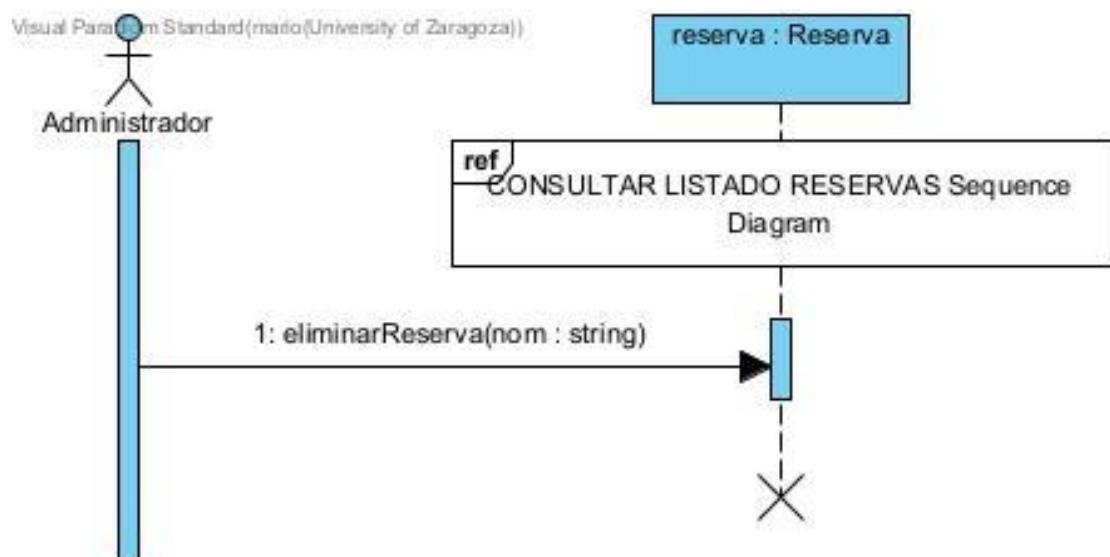
### → Enviar información reserva



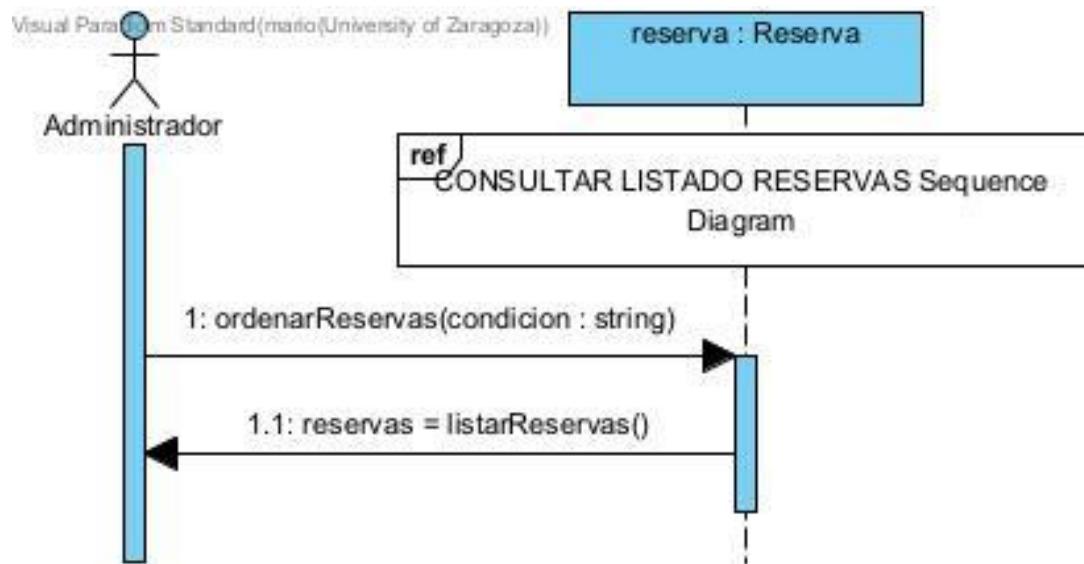
### → Consultar listado reservas



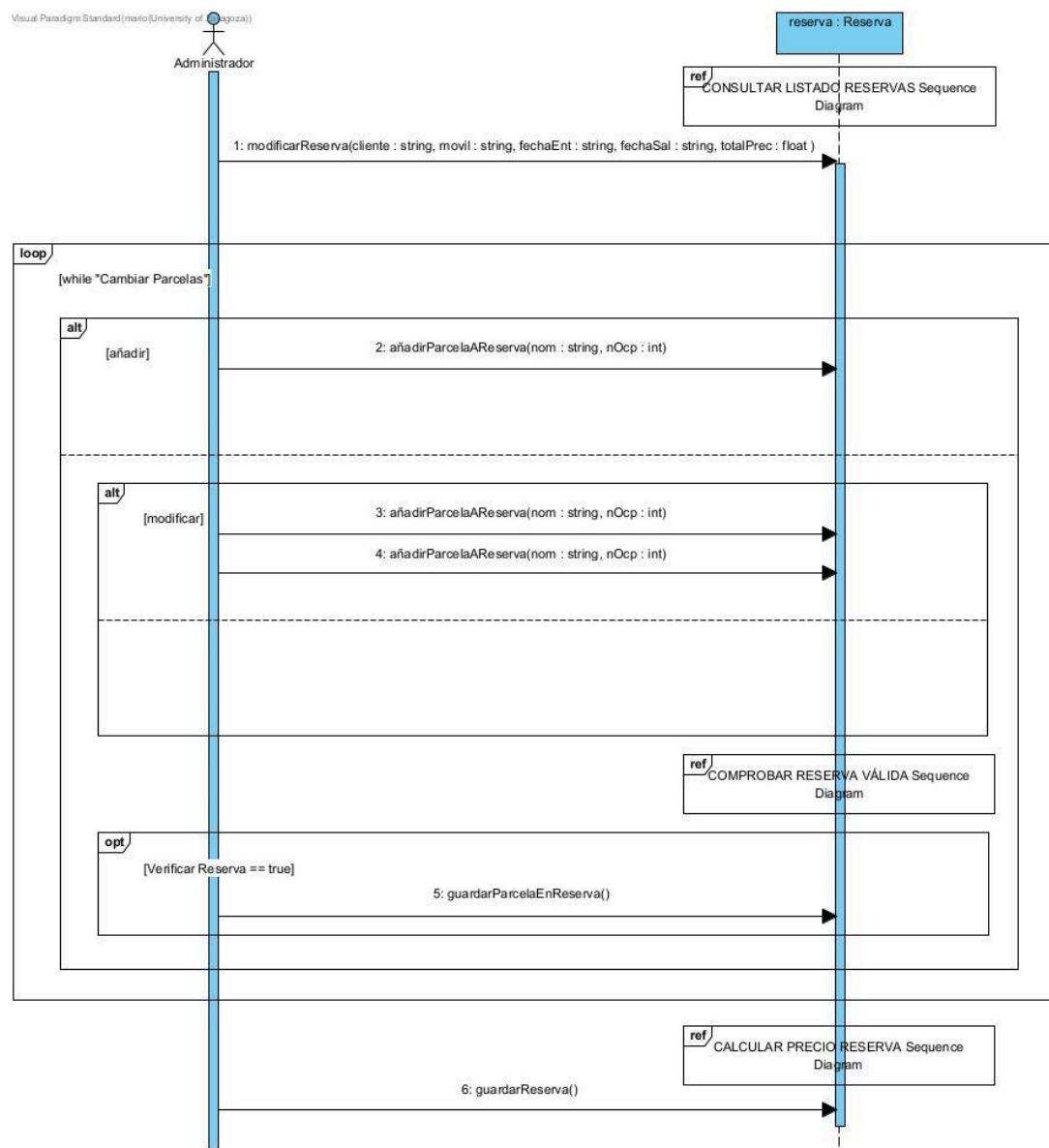
### → Eliminar reserva



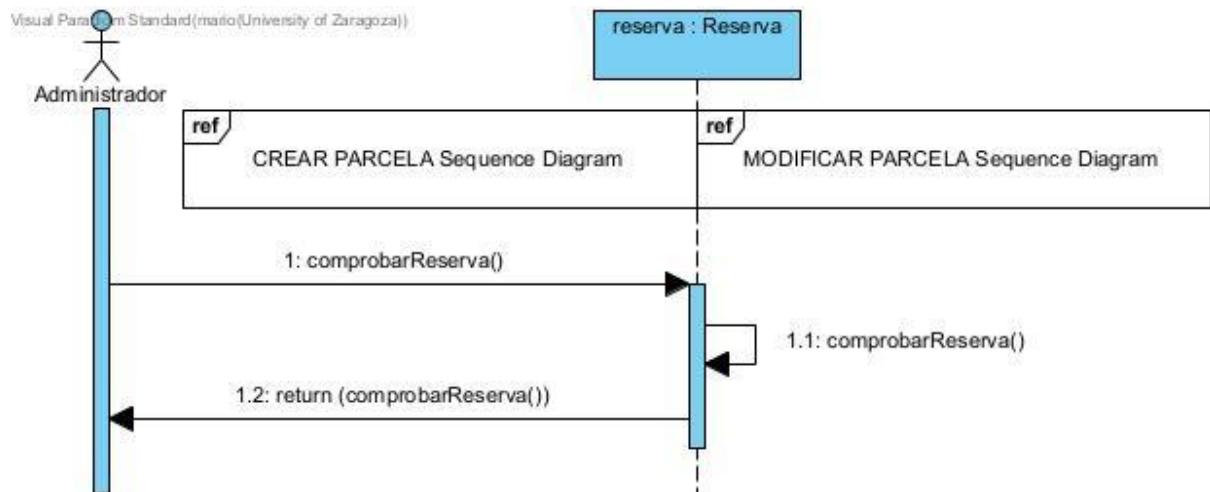
## → Ordenar reserva



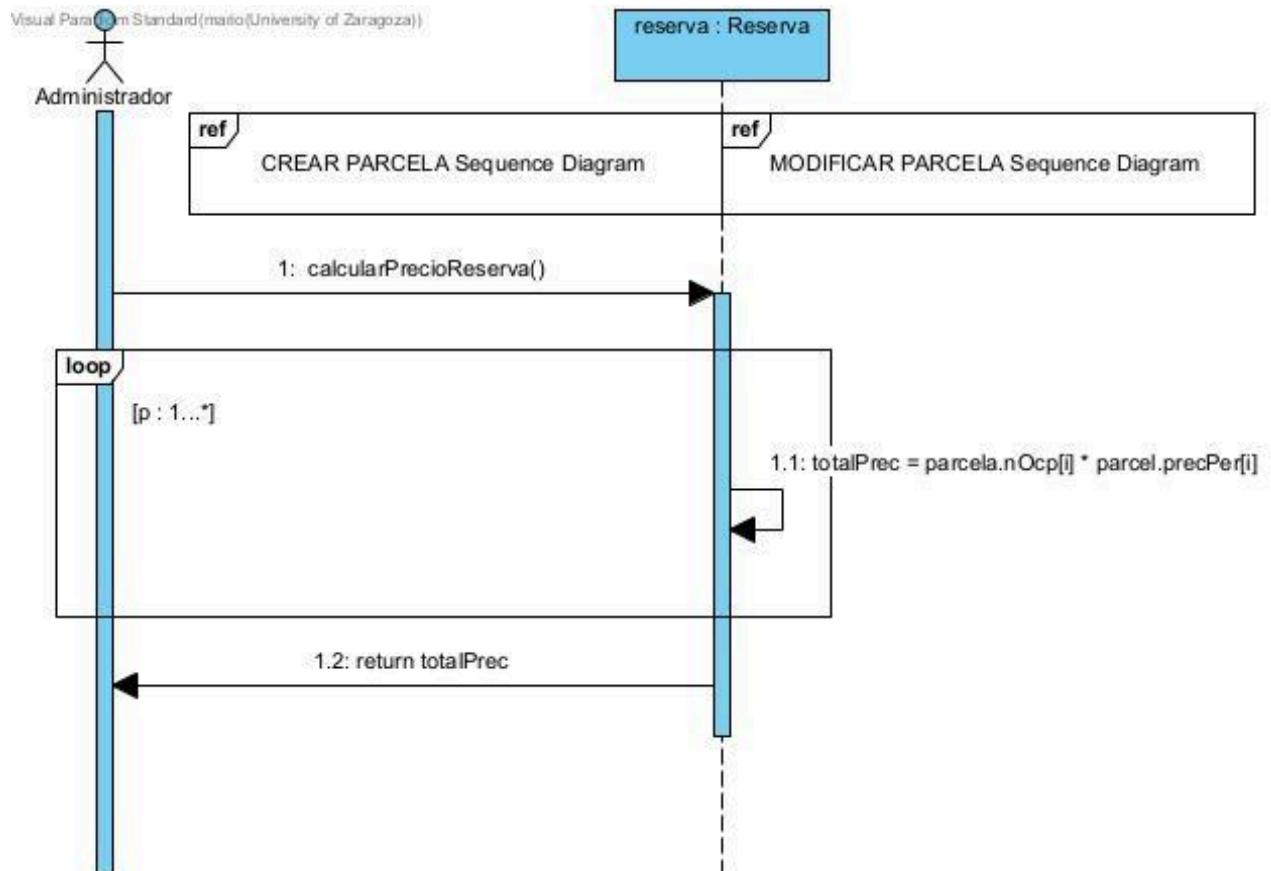
## → Modificar reserva



## → Comprobar reserva válida



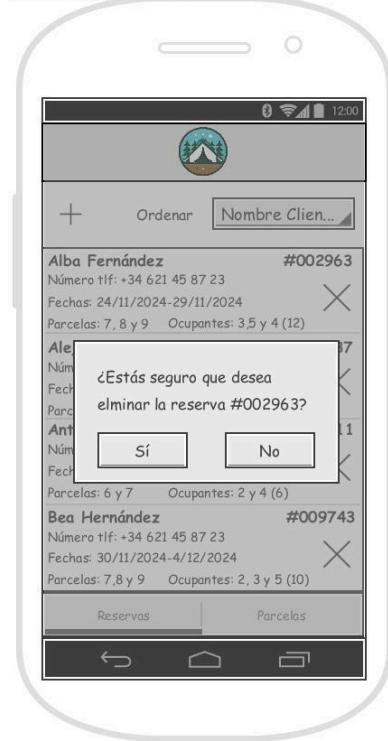
## → Calcular precio reserva



# 6. Prototipado De La Interfaz De Usuario

## Prototipado De Pantallas





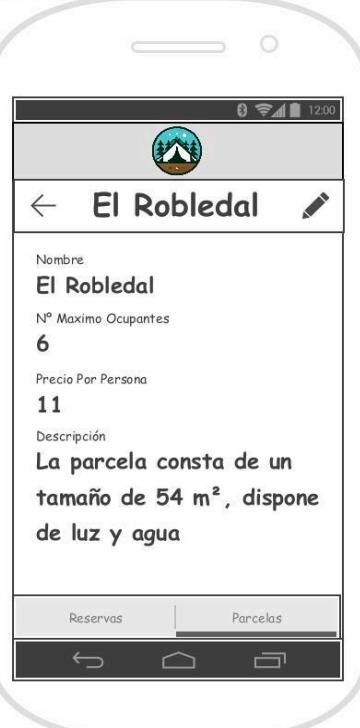
5.



6.



7.



8.

igm Standard(mario)(University of Zaragoza)

#002963

Nombre y apellidos\*  
Alba Fernández

Teléfono\*  
+34 621 45 87 23

Fecha Entrada\* 24/11/2024      Fecha Salida\* 29/11/2024

Parcelas\* + \* Campos obligatorios

× Parcela "El Robledal" 3/6 < >

× Parcela "Los Pinos" 5/9 < >

No hay mas parcelas

Reservas | Parcelas

9.

igm Standard(mario)(University of Zaragoza)

#002963

Nombre y apellidos  
Alba Fernández

Teléfono  
+34 621 45 87 23

Fecha Entrada 24/11/2024      Fecha Salida 29/11/2024

Parcelas +

× Parcela "El Robledal" 3/6 < >

× Parcela "Los Pinos" 5/9 < >

Parcela "Las Brisas" 0/7 < >

Reservas | Parcelas

10.

Visual Paradigm Standard(mario)(University of Zaragoza)

← El Robledal

Nombre\*  
El Robledal

Nº Maximo Ocupantes\*  
6

Precio Por Persona\*  
11

Descripción\*

\* Campos obligatorios

Reservas | Parcelas

11.

igm Standard(mario)(University of Zaragoza)

← Nueva

Nombre y apellidos\*

Teléfono\*

Fecha Entrada\* dd/mm/aaaa      Fecha Salida\* dd/mm/aaaa

Parcelas\* + \* Campos obligatorios

No hay mas parcelas

Reservas | Parcelas

12.

The image displays two side-by-side screenshots of a mobile application interface, labeled 13. and 14., illustrating the process of creating a new reservation.

**Screenshot 13. (Left):** This screen shows the form for creating a new reservation. At the top is a logo of a tent inside a circle. Below it, the word "Nueva" is displayed in a large, bold font, with a back arrow to its left and a save icon to its right. The form contains four input fields with asterisks indicating they are required: "Nombre\*" (Name), "Nº Maximo Ocupantes\*" (Maximum Occupants), "Precio Por Persona\*" (Price per Person), and "Descripción\*" (Description). Below these fields is a note: "\* Campos obligatorios". At the bottom of the form are two buttons: "Reservas" and "Parcelas", with "Reservas" being the active tab. Navigation icons for back, forward, and home are at the very bottom.

**Screenshot 14. (Right):** This screen shows the confirmation of a new reservation. At the top is a logo of a tent inside a circle. The reservation number "#002963" is prominently displayed, with a back arrow to its left and a pencil icon to its right. Below the number, the client's name "Alba Fernández" is shown with arrows indicating it can be edited. A box titled "Información de la reserva enviada al teléfono móvil del cliente" (Reservation information sent to the client's mobile phone) contains the message: "Parcela \"El Robledal\" 3/6", "Parcela \"Los Pinos\" 5/9", and "Parcela \"Las Brisas\"". At the bottom, there are tabs for "Reservas" and "Parcelas", with "Reservas" being the active tab. Navigation icons for back, forward, and home are at the very bottom.

13.

14.

# Mapa De Navegación Del Prototipo



## Matriz Relacional Casos De Uso con Pantallas

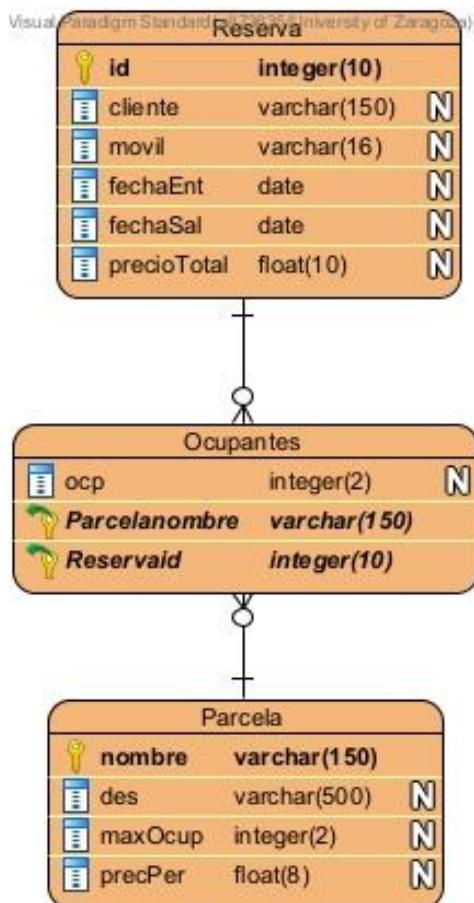
La matriz relacional establece una correspondencia directa entre los casos de uso y las pantallas del prototipo definidas en el apartado anterior, con ella se pueden observar más fácilmente que pantallas cubren cada caso de uso.

Casos de Uso \ Pantallas	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
<b>1</b>				X										X
<b>2</b>				X										
<b>3</b>				X		X								
<b>4</b>				X				X				X		
<b>5</b>				X										
<b>6</b>			X				X							X
<b>7</b>			X										X	
<b>8</b>			X											
<b>9</b>			X				X		X	X				
<b>10</b>														
<b>11</b>														
<b>12</b>					X									
<b>13</b>			X											

# 7. Diseño del Sistema y Diseño de Objetos

## Modelo lógico de la base de datos relacional

Se ha diseñado el esquema entidad-relación para gestionar las reservas del camping. Además, se han definido las sentencias SQL tanto para la creación como destrucción de las tablas.



### Creación de tablas:

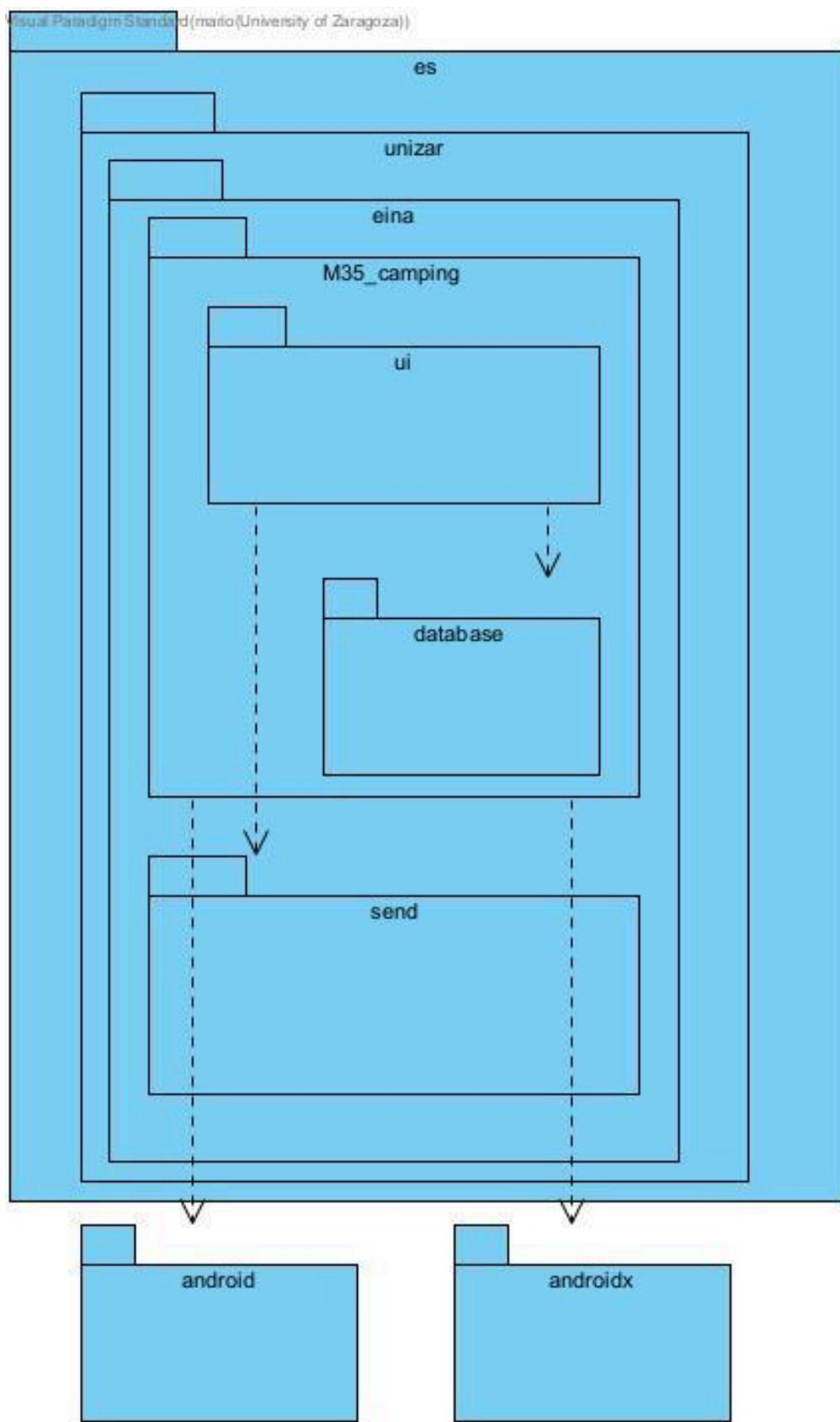
```
CREATE TABLE Reserva (
    id INTEGER PRIMARY KEY,
    cliente TEXT NOT NULL,
    movil TEXT NOT NULL,
    fechaEnt DATE NOT NULL,
    fechaSal DATE NOT NULL,
    precioTotal      REAL      NOT      NULL
    CHECK(precioTotal>0)
);
CREATE TABLE Parcela (
    nombre TEXT PRIMARY KEY,
    des TEXT NOT NULL,
    maxOcup INTEGER NOT NULL CHECK(maxOcup>0),
    precPer REAL NOT NULL CHECK(precPer>0)
);
CREATE TABLE Ocupantes (
    ocp INTEGER NOT NULL CHECK(ocp>0),
    Parcelanombre      TEXT      REFERENCES
    Parcela(nombre) ON DELETE CASCADE,
    Reservaid INTEGER REFERENCES Reserva(id)
    ON DELETE CASCADE,
    PRIMARY KEY(Parcelanombre,Reservaid)
);
```

### Destrucción de tablas:

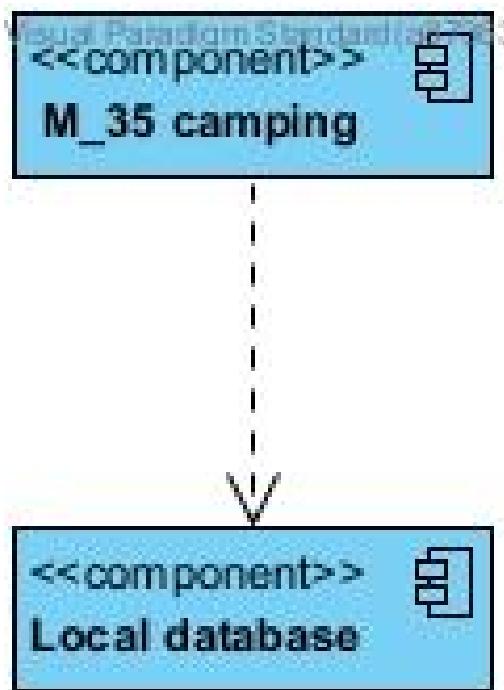
```
-- Primero se elimina la tabla dependiente
DROP TABLE IF EXISTS Ocupantes;
```

```
-- Después las tablas independientes
DROP TABLE IF EXISTS Reserva;
DROP TABLE IF EXISTS Parcela;
```

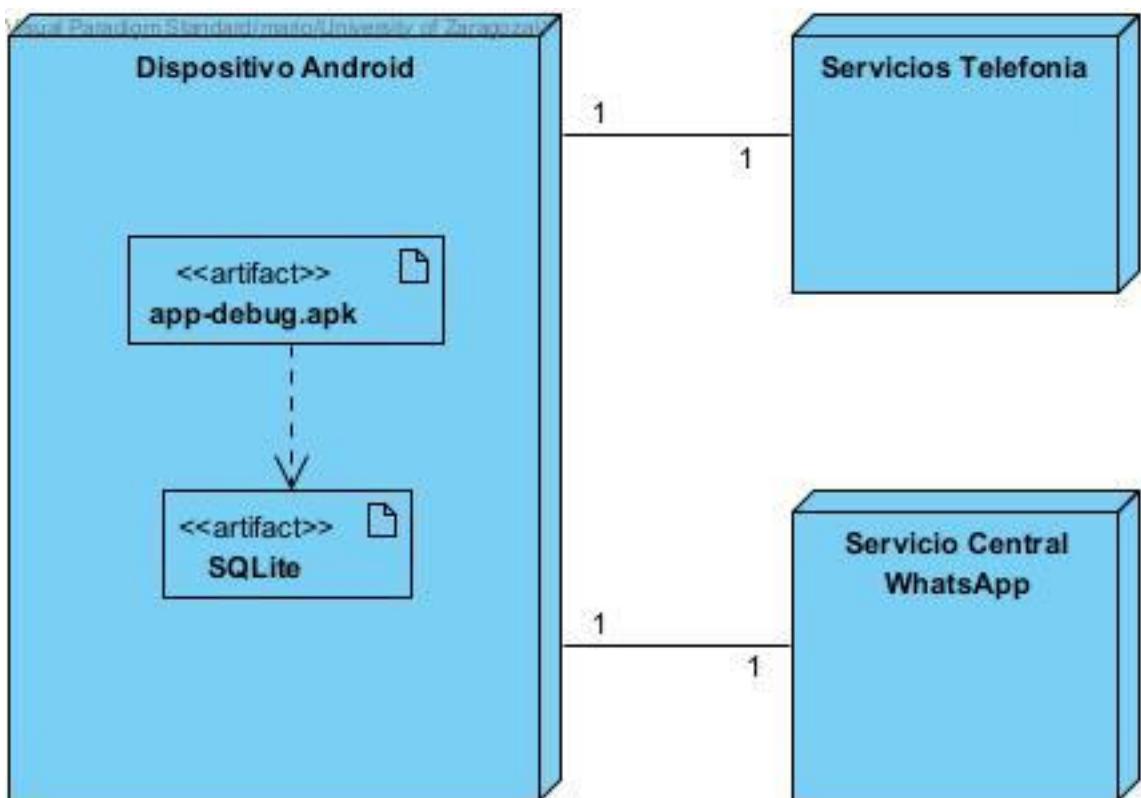
## Diagrama de paquetes



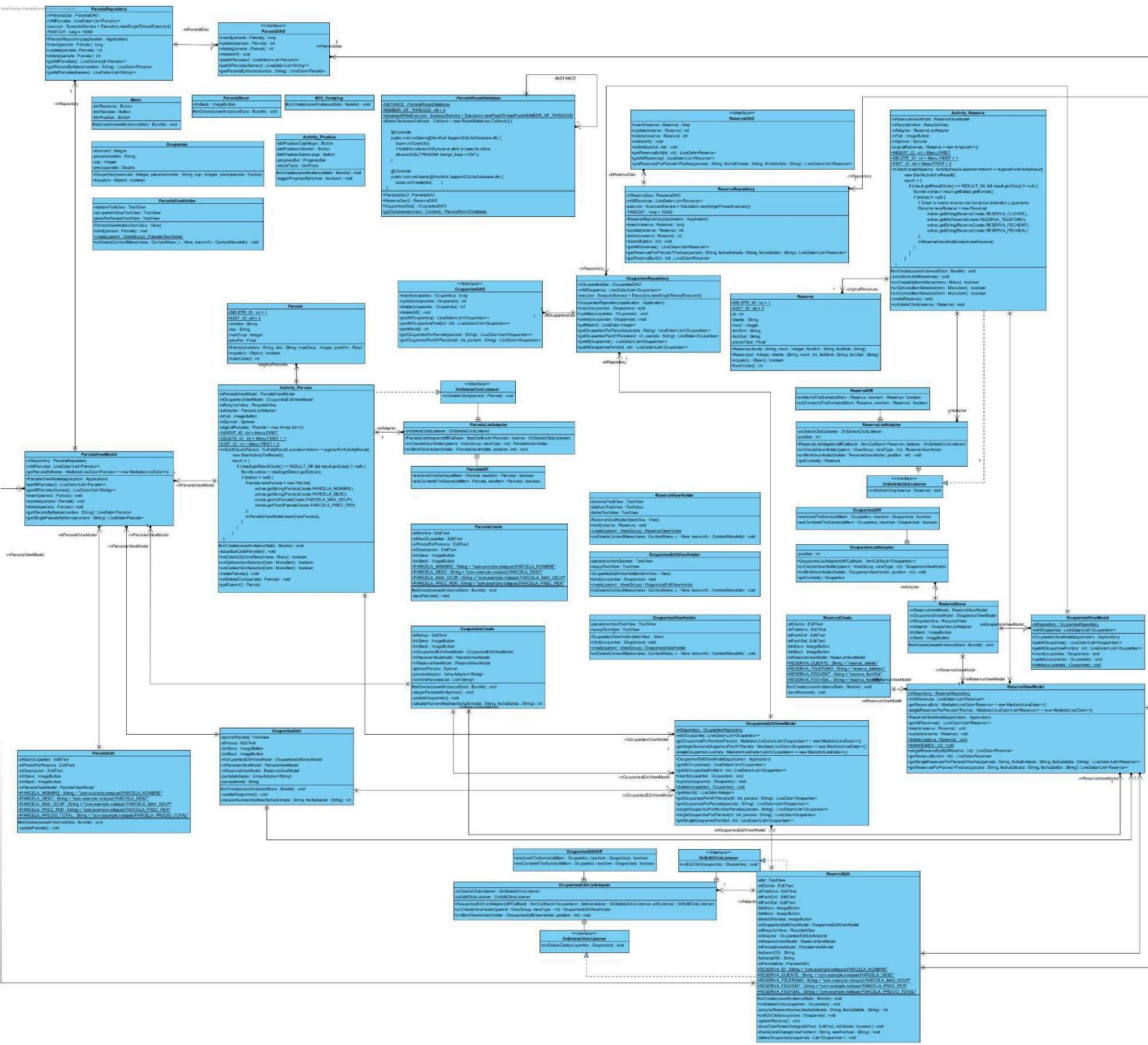
## Diagrama de componentes



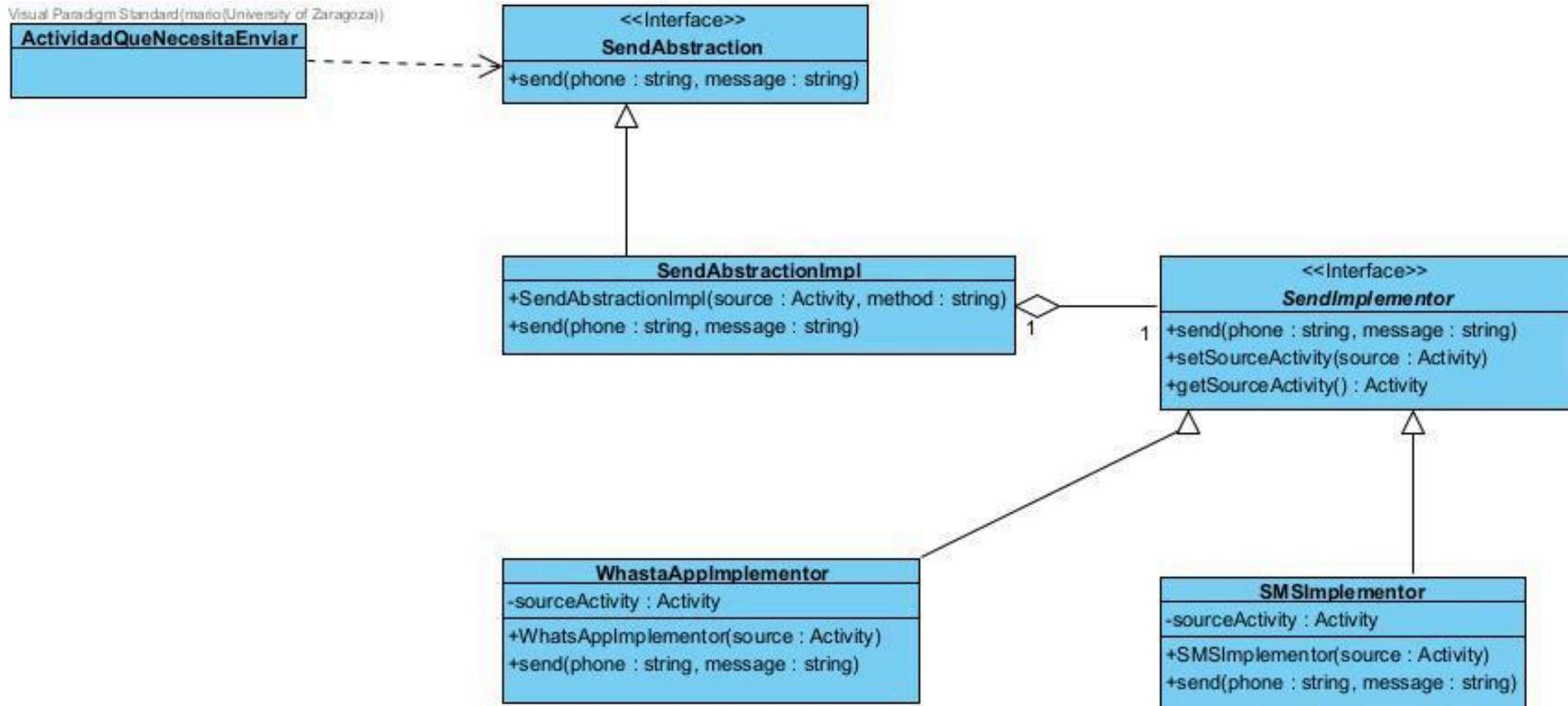
## Diagrama de despliegue



## Diagrama de clases

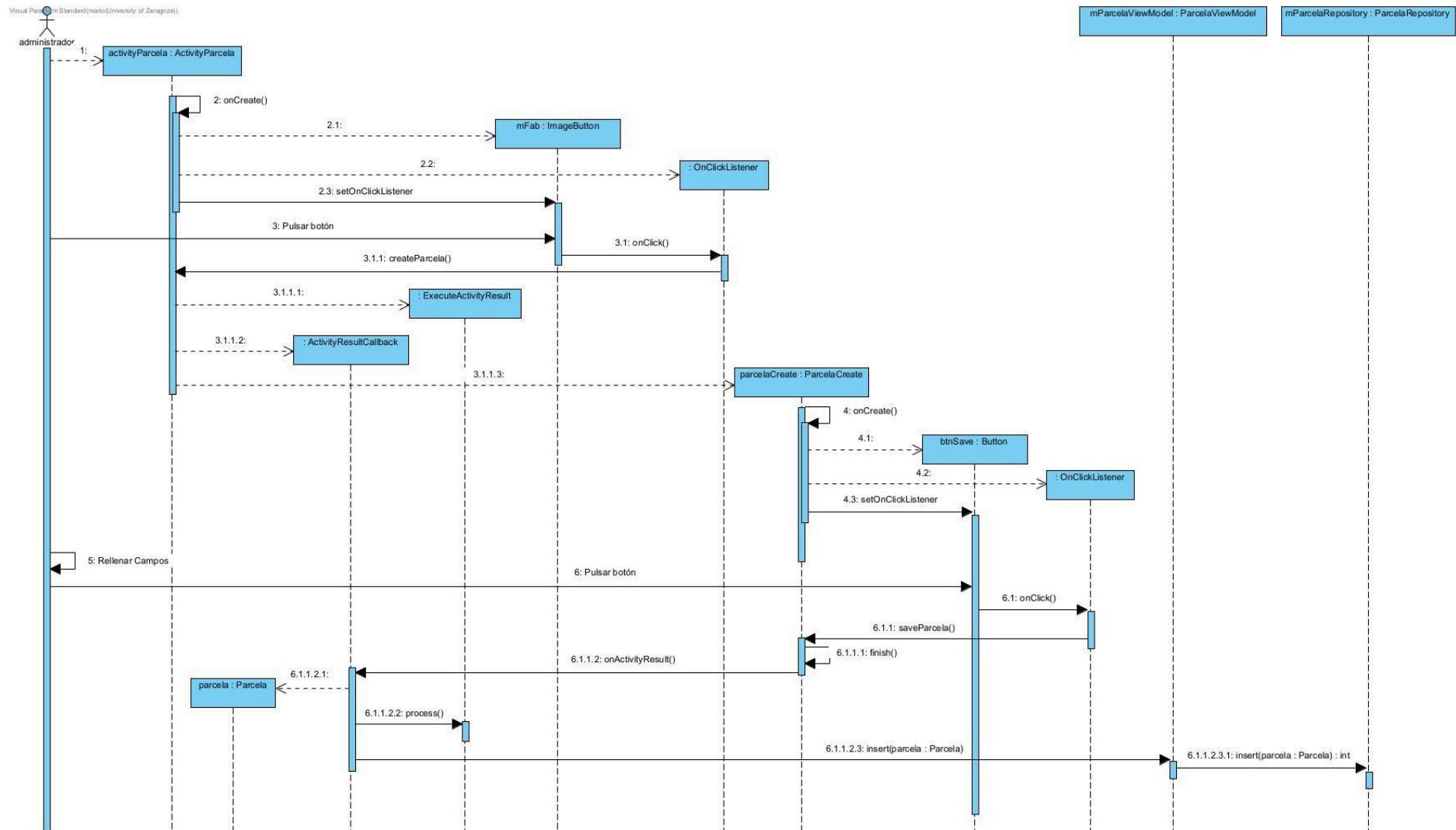


## Diagrama de clases Patrón Bridge

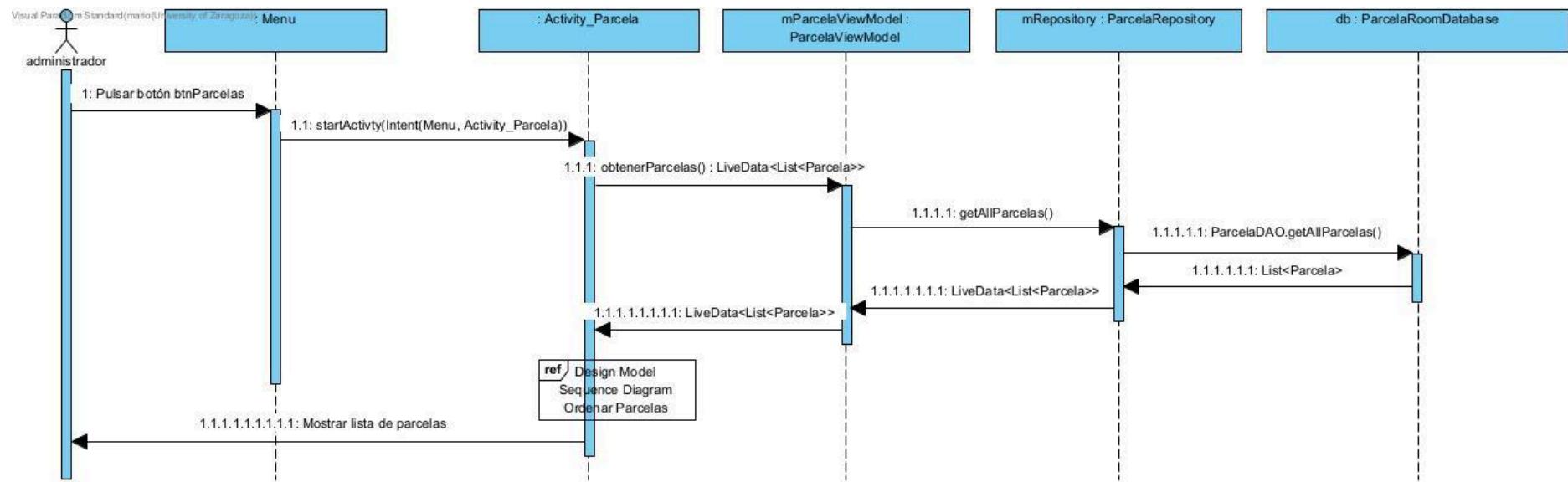


## Diagramas de secuencia a nivel de diseño

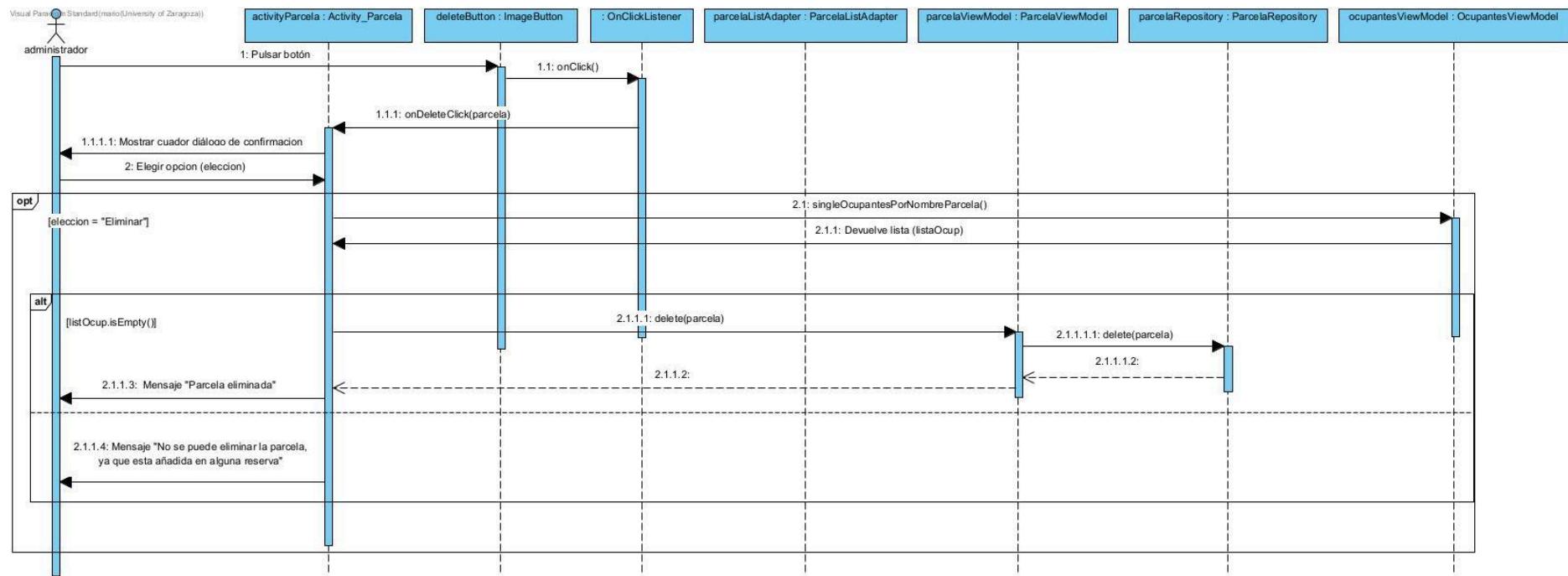
### → Crear parcela



## → Consultar parcelas

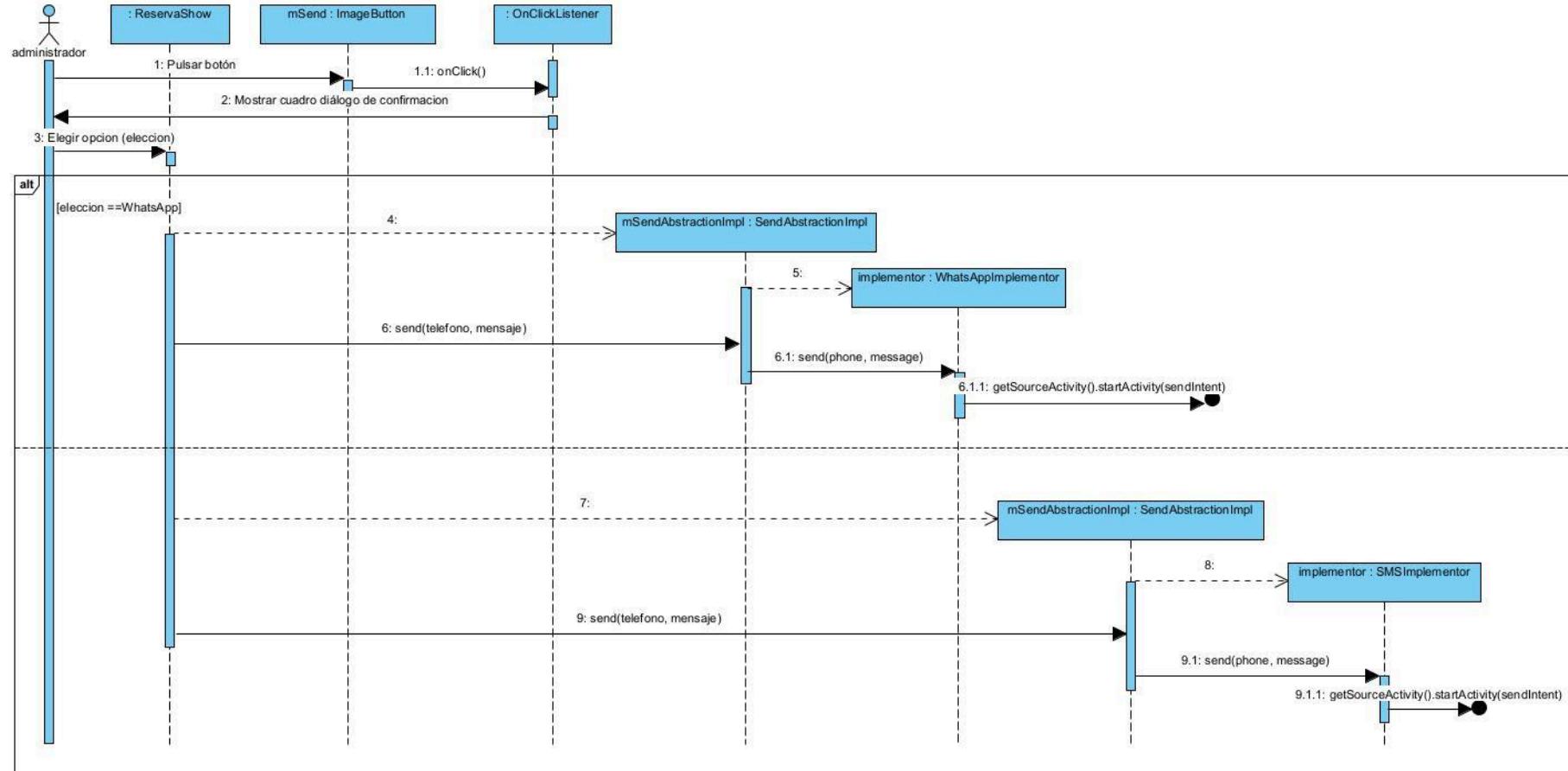


## → Eliminar parcela

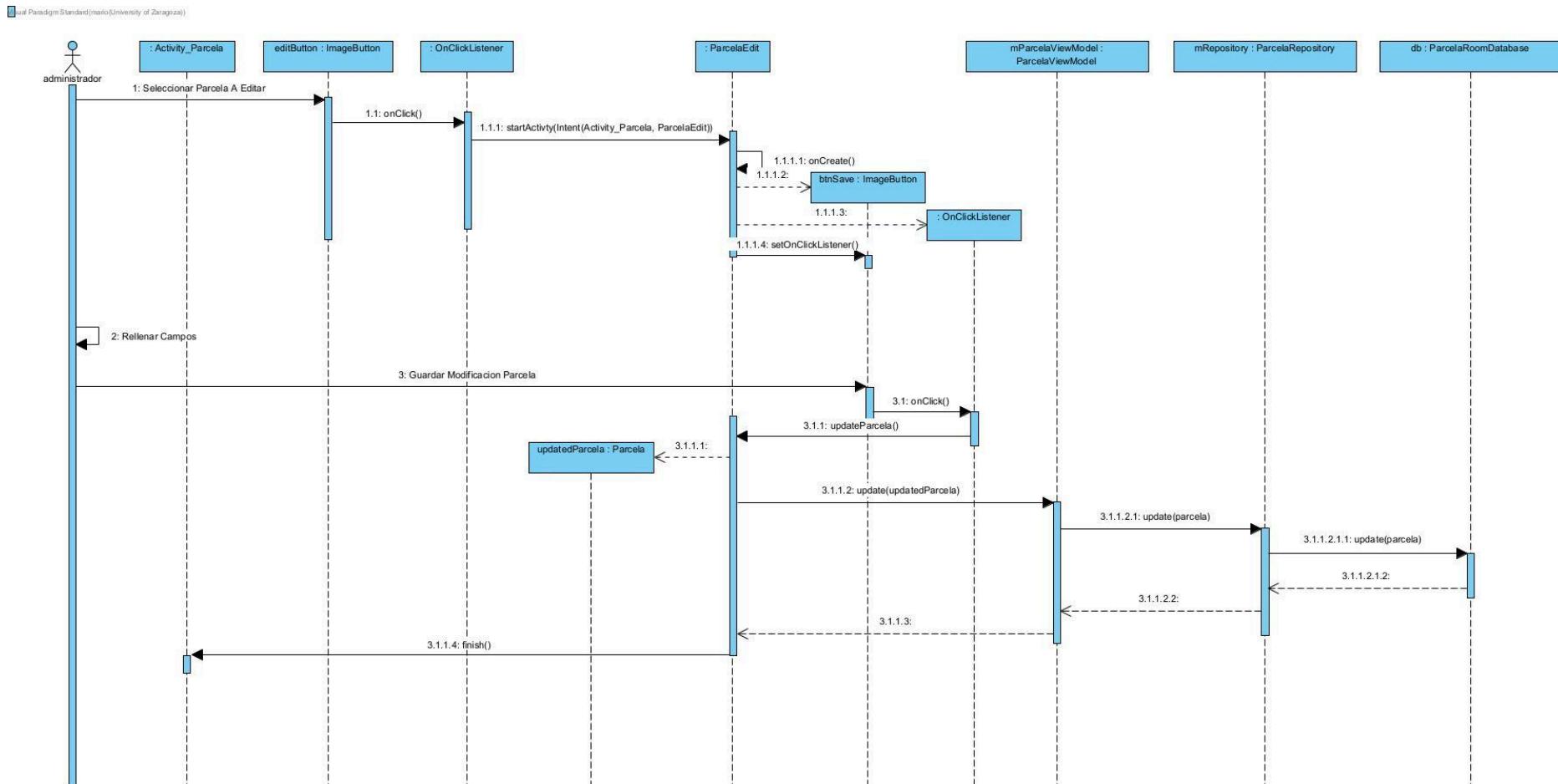


## → Enviar información reserva

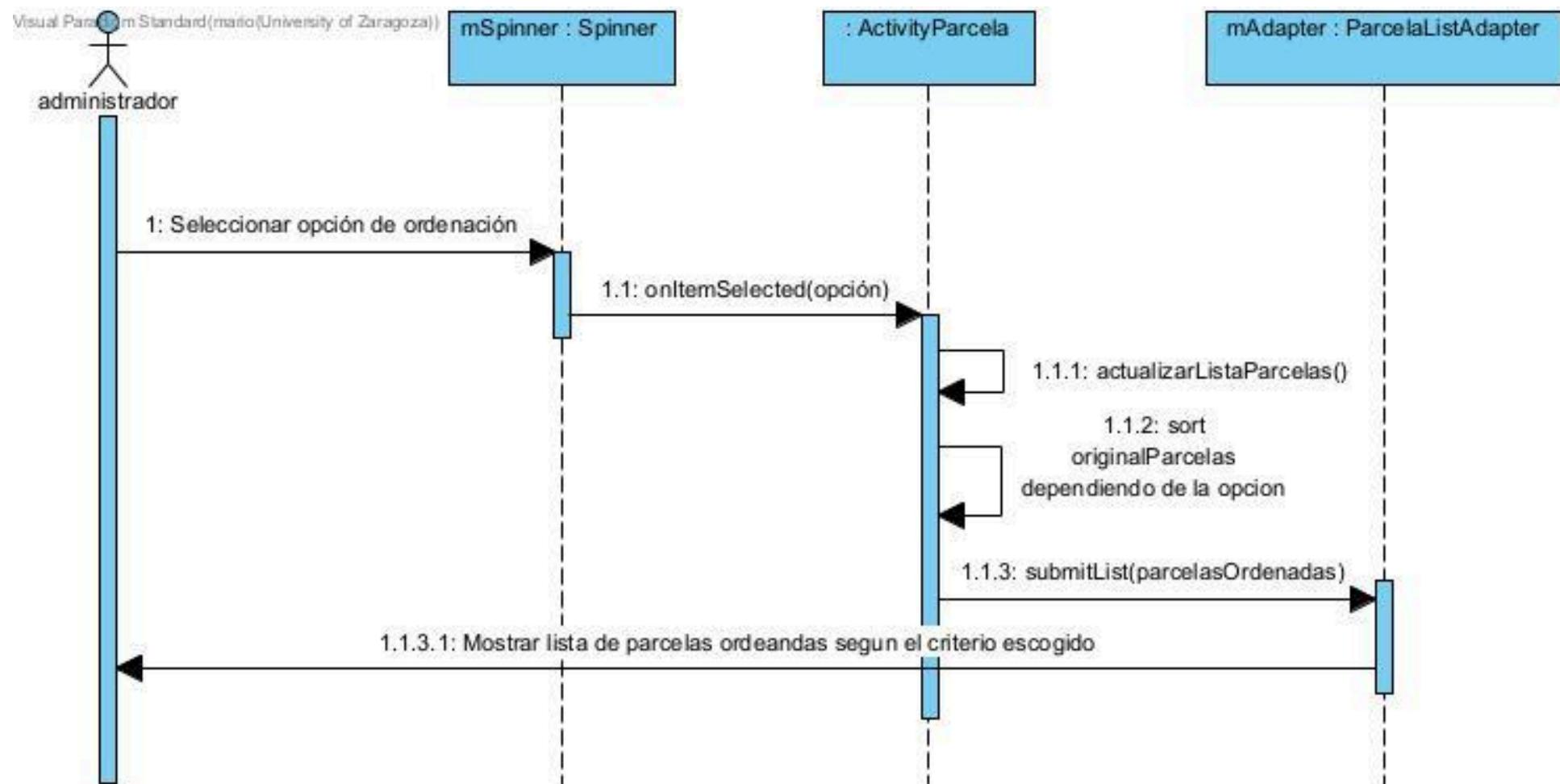
UML Paradigm Standard(mario)(University of Zaragoza)



## → Modificar parcela

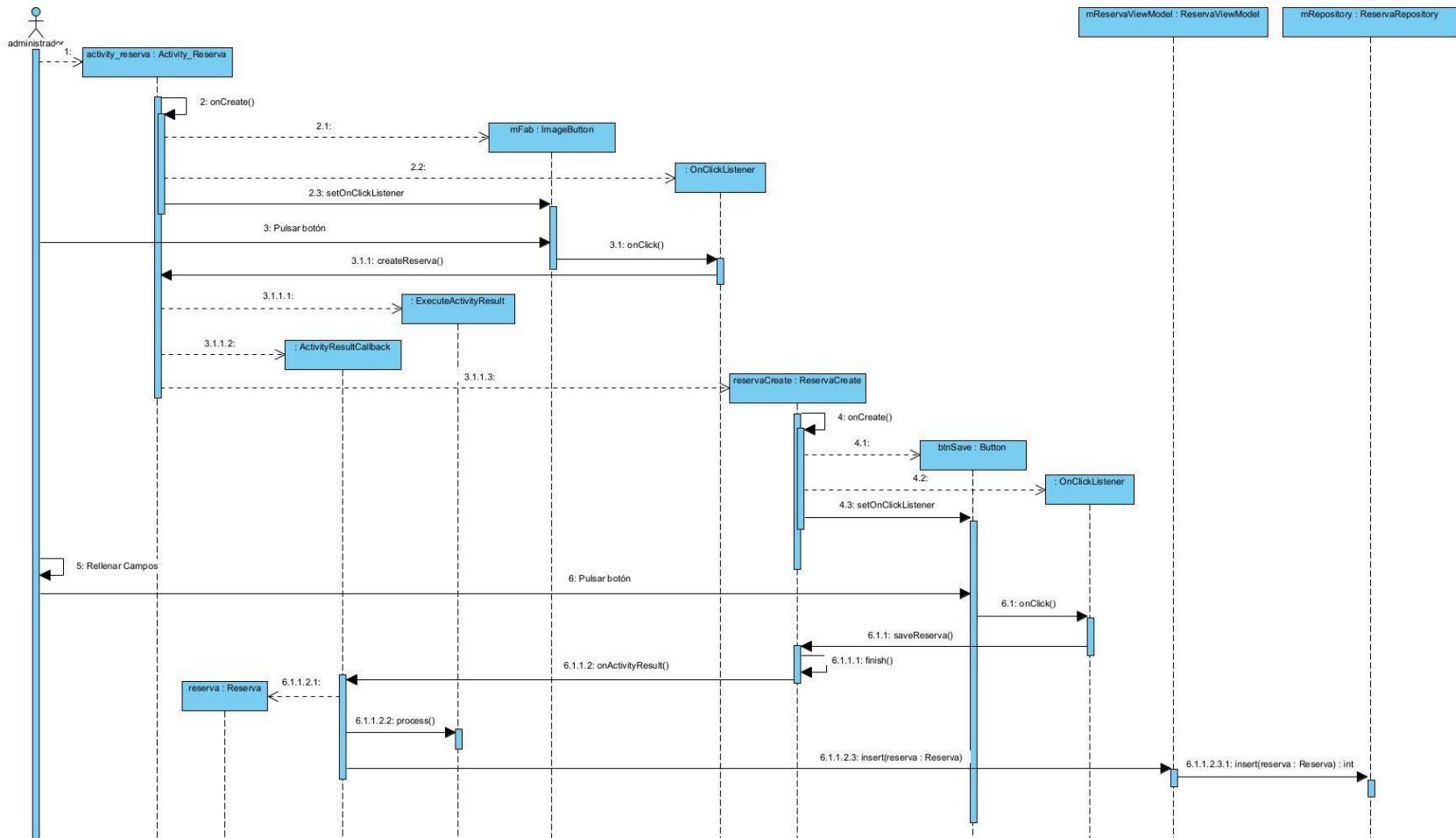


## → Ordenar parcela

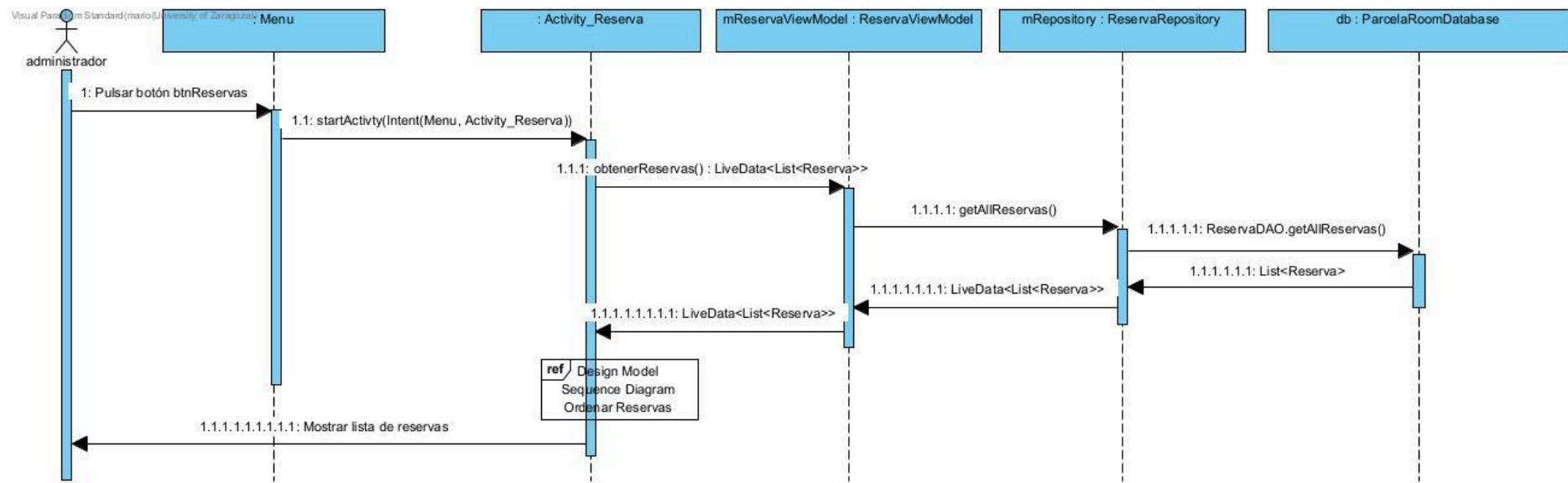


## → Crear reserva

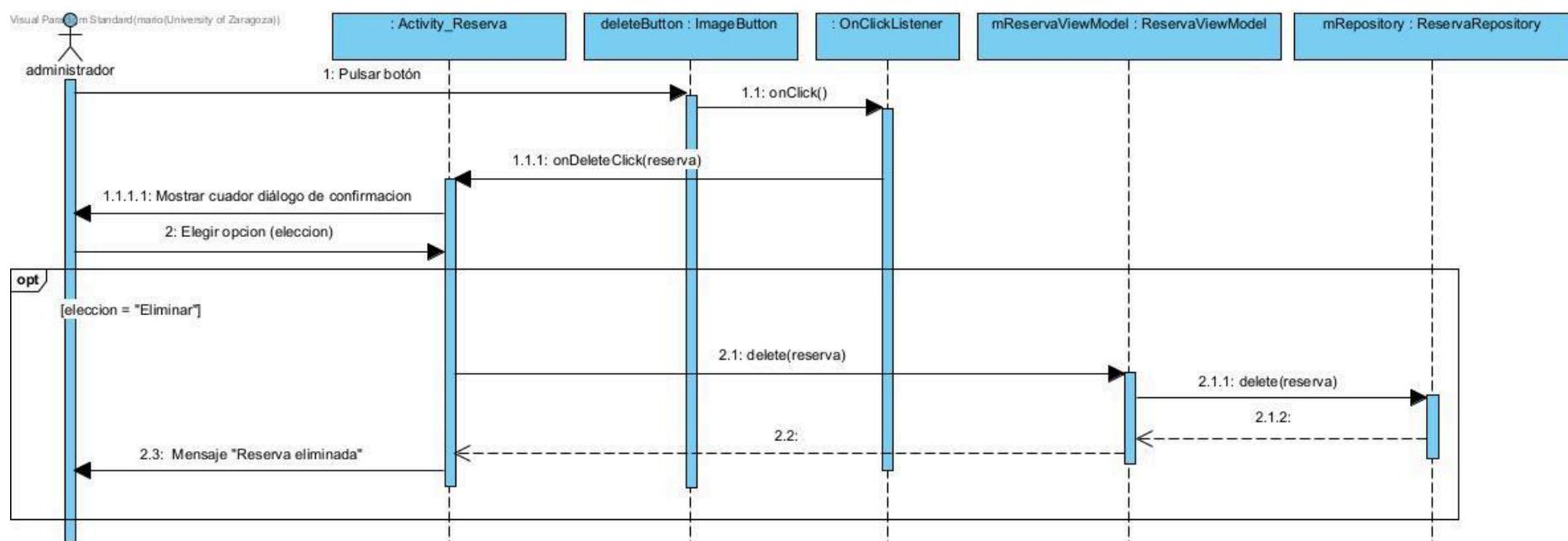
UML Paradigm Standard(malo)(University of Zaragoza)



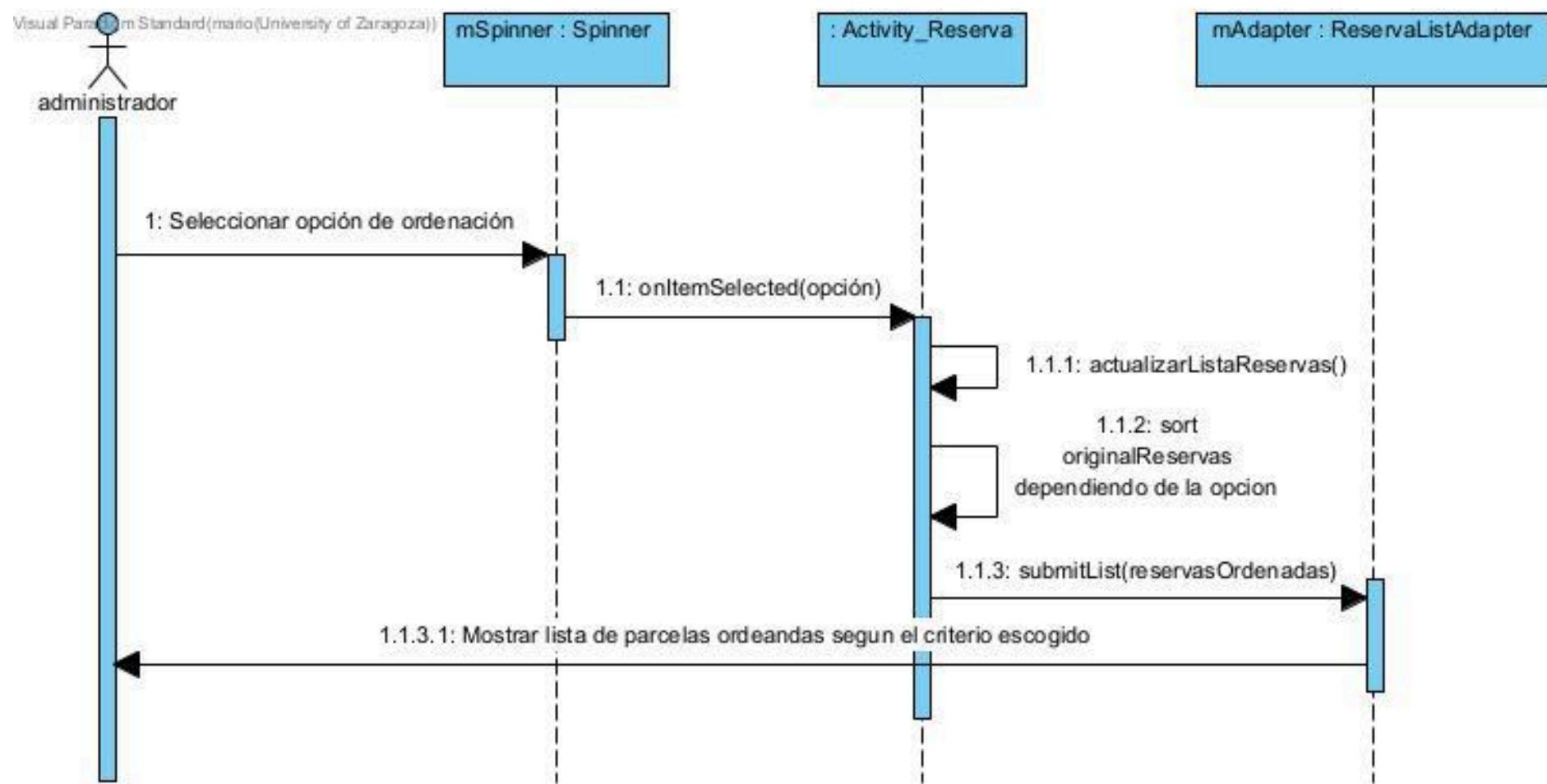
## → Consultar listado reservas



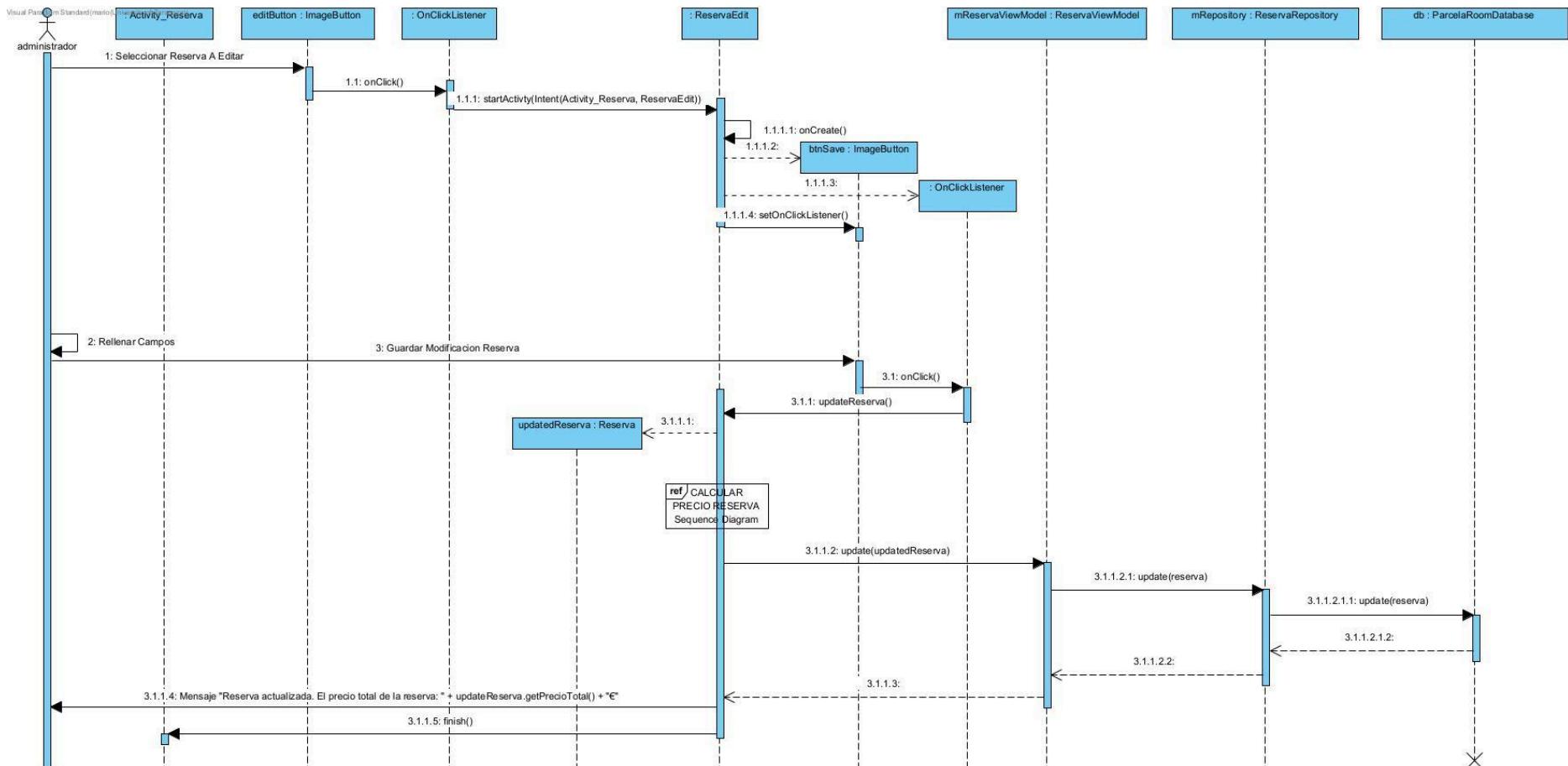
## → Eliminar reserva



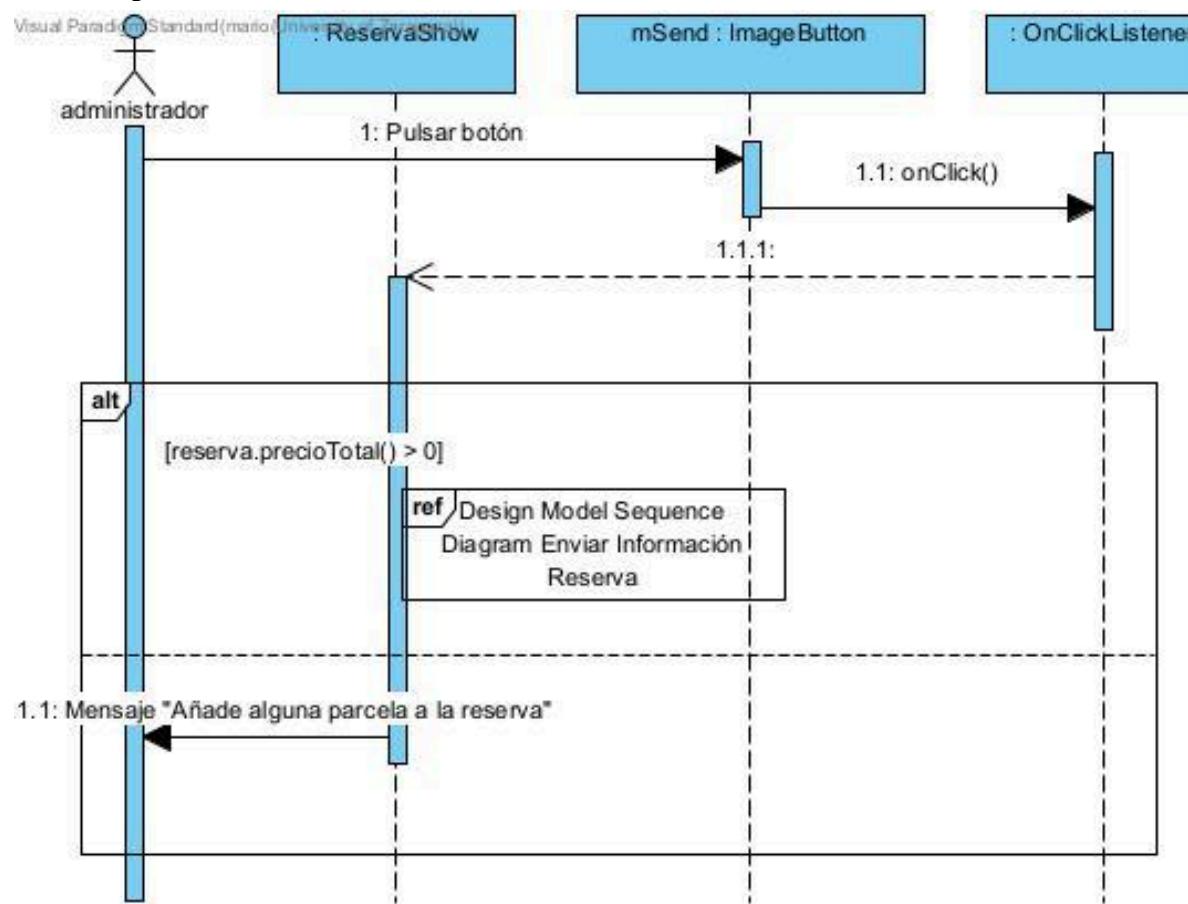
## → Ordenar reserva



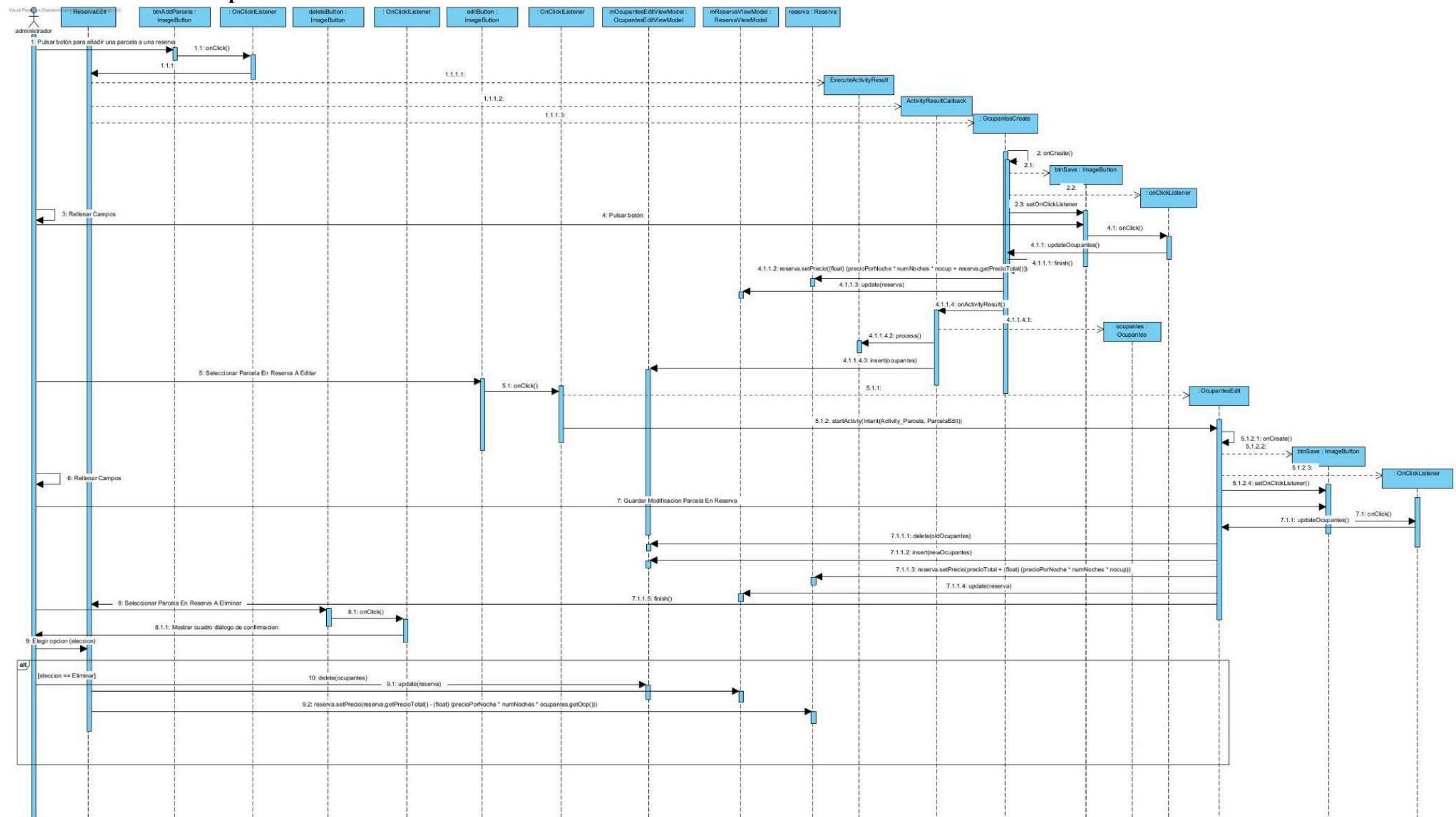
## → Modificar reserva



## → Comprobar reserva válida



## → Calcular precio reserva



# 8. Implementación

## Prototipado De Pantallas Finales

1. 

Campify®
2. 

¿A dónde quieres acceder?

PRUEBAS

RESERVAS PARCELAS
3. 

Marta Ibañez Alloza #1  
Número tlf: 601400488  
Fechas: 2024-01-01 al 2024-11-14

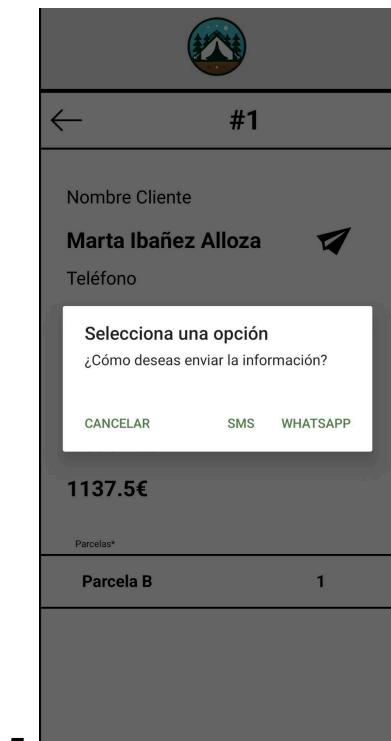
Jorge Hernández Aznar #2  
Número tlf: 653745457  
Fechas: 2024-02-10 al 2024-03-15
4. 

Nombre Cliente  
**Marta Ibañez Alloza** 

Teléfono  
**601400488**

Fecha Entrada Fecha Salida  
**2024-01-01 2024-11-14**

Precio Total  
**0.0€**

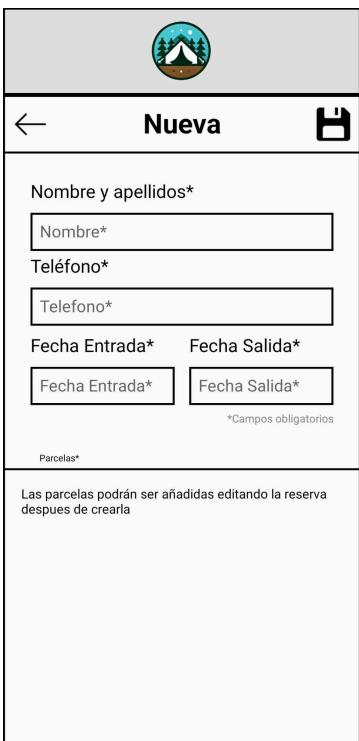
Parcelas\*
5. 

Selección una opción  
¿Cómo deseas enviar la información?

CANCELAR SMS WHATSAPP

1137.5€

Parcelas\*

Parcela B 1
6. 

Nueva

Nombre y apellidos\*

Teléfono\*

Fecha Entrada\* Fecha Salida\*

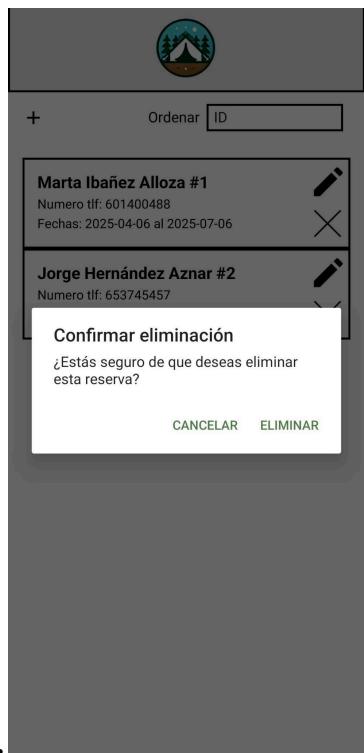
Fecha Entrada\* Fecha Salida\*

\*Campos obligatorios

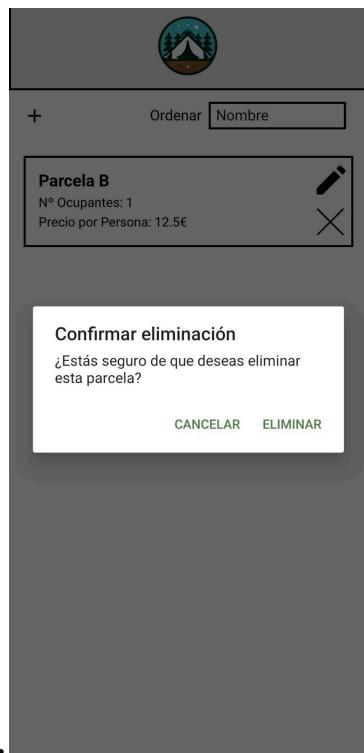
Parcelas\*

Las parcelas podrán ser añadidas editando la reserva después de crearla

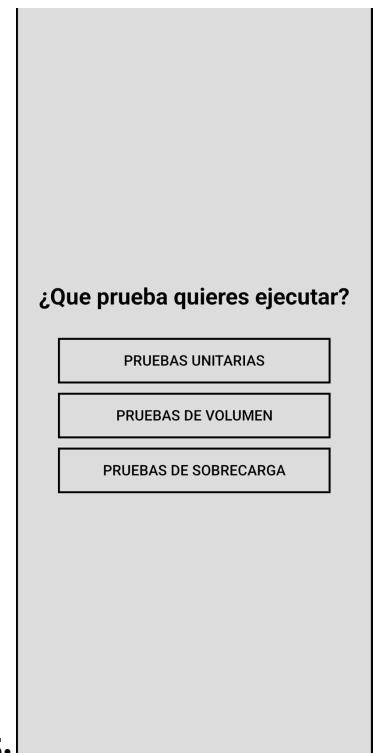
<p><b>7.</b></p>  <p>← #1 </p> <p>Nombre y apellidos* Marta Ibañez Alloza</p> <p>Teléfono* 601400488</p> <p>Fecha Entrada* Fecha Salida* 2025-04-06 2025-07-06</p> <p>* Campos obligatorios</p> <p>Parcelas* +  <b>Parcela B</b> 1 </p>	<p><b>8.</b></p>  <p>← Nueva </p> <p>Parcela* <input type="text" value="Parcela B"/></p> <p>Nº Ocupantes* <input type="text" value="Nº Maximo Ocupantes*"/></p> <p>* Campos obligatorios</p>	<p><b>9.</b></p>  <p>← Nueva </p> <p>Parcela Parcela B</p> <p>Nº Ocupantes* <input type="text" value="1"/></p> <p>* Campos obligatorios</p>
<p><b>10.</b></p>  <p>+ Ordenar <input type="text" value="Nombre"/></p> <p><b>Parcela B</b> Nº Ocupantes: 1 Precio por Persona: 12.5€ </p>	<p><b>11.</b></p>  <p>← Nueva </p> <p>Nombre* <input type="text" value="Nombre*"/></p> <p>Nº Maximo Ocupantes* <input type="text" value="0"/></p> <p>Precio Por Persona* <input type="text" value="0.0"/></p> <p>Descripción* <input type="text" value="Descripción*"/></p> <p>* Campos obligatorios</p>	<p><b>12.</b></p>  <p>← Parcela B </p> <p>Nº Maximo Ocupantes* <input type="text" value="1"/></p> <p>Precio Por Persona* <input type="text" value="12.5"/></p> <p>Descripción* <input type="text" value="Descripción B"/></p> <p>* Campos obligatorios</p>



13.

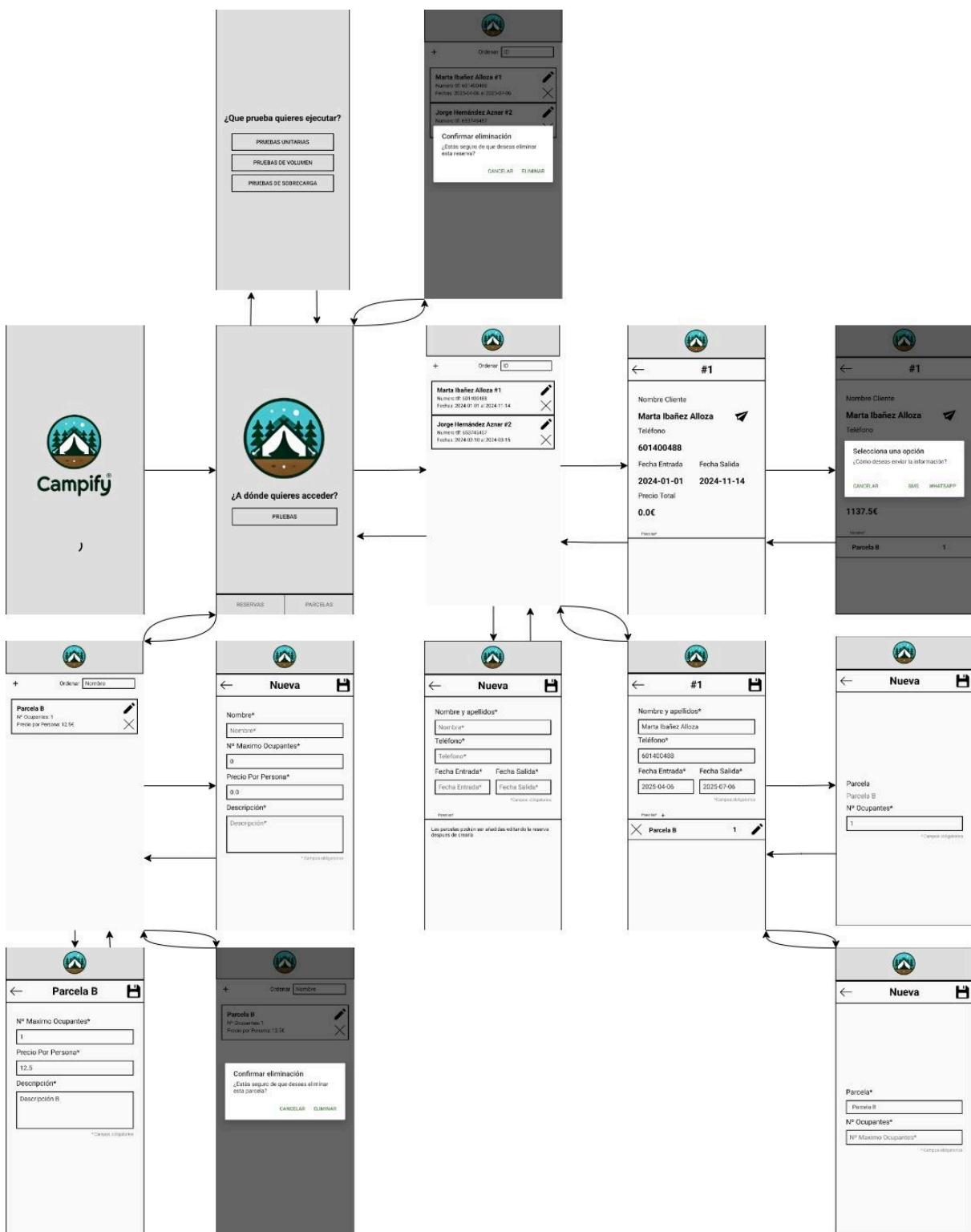


14.



15.

## Mapa De Navegación Final



## 9. Pruebas

El desarrollo y mantenimiento de aplicaciones requiere asegurar que estas cumplan con los requisitos funcionales establecidos y que se comporten correctamente bajo diversas condiciones. Para ello, es fundamental realizar un proceso exhaustivo de pruebas que evalúe tanto su correcto funcionamiento como su capacidad para manejar datos inusuales o límites extremos. En este caso, se han diseñado una serie de pruebas para las funcionalidades relacionadas con la gestión de parcelas y reservas en la aplicación.

El objetivo de estas pruebas es garantizar que los datos introducidos en la aplicación sean válidos, evitando entradas nulas o con formatos incorrectos. Además, se busca asegurar que los límites de la aplicación sean correctamente manejados, como el máximo de caracteres en descripciones, valores numéricos o el manejo de grandes volúmenes de datos. Asimismo, es importante que los errores sean detectados de manera adecuada cuando se introducen datos inválidos. Estas verificaciones son esenciales para mantener la integridad y la robustez del sistema.

### Pruebas unitarias de caja negra

Estas pruebas evalúan que las operaciones principales, como insertar, actualizar y eliminar parcelas o reservas, se comporten adecuadamente bajo diferentes escenarios. Para lograr estos objetivos, se han diseñado clases de equivalencia que definen parámetros válidos e inválidos para los métodos de las clases `ParcelaRepository` y `ReservaRepository`. Estas pruebas incluyen condiciones de entrada esperadas, como nombres no nulos y fechas con formato válido, así como entradas inválidas, como valores nulos, formatos erróneos o límites fuera de rango.

Por último, se han establecido casos de prueba específicos para evaluar los métodos principales tanto para las parcelas como para las reservas. A continuación, se presentan las tablas con las clases de equivalencia y los casos de prueba diseñados para validar el comportamiento esperado de las operaciones principales.

#### Clases de equivalencia de la clase `ParcelaRepository`:

Parámetros	Clase de equivalencia válida	Clase de equivalencia no válida
<code>nombre</code>	1) nombre != null 2) nombre.length() > 0	9) nombre == null 10) nombre.length() == 0
<code>des</code>	3) des != null 4) des.length() > 0	11) des == null 12) des.length() == 0
<code>maxOcup</code>	5) maxOcup != null 6) maxOcup > 0	13) maxOcup == null 14) maxOcup == 0
<code>precPer</code>	7) precPer != null 8) precPer > 0.0	15) parcela.precPer == null 16) parcela.precPer == 0.0

### Casos de prueba para el método insert de la clase ParcelaRepository

Caso	nombre	des	maxOcup	precPer	Resultado esperado	Clases Cubiertas
<b>Casos de prueba para las clases de equivalencia válidas</b>						
1	“título”	“descripcion”	4	14.0	> 0	1,2,3,4,5,6,7,8
<b>Casos de prueba para las clases de equivalencia no válidas</b>						
2	null	“descripcion”	4	14.0	-1	9
3	“”	“descripcion”	4	14.0	-1	10
4	“título”	null	4	14.0	-1	11
5	“título”	“”	4	14.0	-1	12
6	“título”	“descripcion”	null	14.0	-1	13
7	“título”	“descripcion”	0	14.0	-1	14
8	“título”	“descripcion”	4	null	-1	15
9	“título”	“descripcion”	4	0.0	-1	16

### Casos de prueba para el método update de la clase ParcelaRepository

Caso	nombre	des	maxOcup	precPer	Resultado esperado	Clases Cubiertas
<b>Casos de prueba para las clases de equivalencia válidas</b>						
1	“título”	“descripcion”	4	14.0	> 0	1,2,3,4,5,6,7,8
<b>Casos de prueba para las clases de equivalencia no válidas</b>						
2	null	“descripcion”	4	14.0	-1	9
3	“”	“descripcion”	4	14.0	-1	10
4	“título”	null	4	14.0	-1	11
5	“título”	“”	4	14.0	-1	12
6	“título”	“descripcion”	null	14.0	-1	13
7	“título”	“descripcion”	0	14.0	-1	14
8	“título”	“descripcion”	4	null	-1	15
9	“título”	“descripcion”	4	0.0	-1	16

## Casos de prueba para el método delete de la clase ParcelaRepository

Caso	nombre	des	maxOcup	precPer	Resultado esperado	Clases Cubiertas
Casos de prueba para las clases de equivalencia válidas						
1	“título”	“descripcion”	4	14.0	True	1,2,3,4,5,6,7,8
Casos de prueba para las clases de equivalencia no válidas						
2	null	“descripcion”	4	14.0	False	9

## Clases de equivalencia de la clase ReservaRepository

Parámetros	Clase de equivalencia válida	Clase de equivalencia no válida
id	1) id >= 0 2) id != null	14) id < 0 15) id == null
cliente	3) cliente != null 4) cliente.length() > 0	16) cliente == null 17) cliente.length() == 0
movil	5) movil > 0	18) movil <= 0
fechEnt	6) tiene formato AAAA-MM-DD 7) fecha del calendario gregoriano 8) fechEnt != null	19) no tiene formato AAAA-MM-DD 20) la fecha no pertenece al calendario gregoriano 21) fechEnt == null
fechSal	9) tiene formato AAAA-MM-DD 10) fecha del calendario gregoriano 11) fechSal != null	22) no tiene formato AAAA-MM-DD 23) la fecha no pertenece al calendario gregoriano 24) fechSal == null
precioTotal	12) precioTotal != null 13) precioTotal >= 0.0	25) precioTotal == null 26) precioTotal < 0.0

## Casos de prueba para el método insert de la clase ReservaRepository

Caso	id	cliente	movil	fechEnt	fechSal	precioTotal	Resultado esperado	Clases Cubiertas
Casos de prueba para las clases de equivalencia válidas								
1	0	“cliente”	654251202	2024-08-12	2024-08-14	24.02	> 1	1,2,3,4,5,6,7,8,9,10,11,12,13
Casos de prueba para las clases de equivalencia no válidas								
2	-1	“cliente”	654251202	2024-08-12	2024-08-14	24.02	-1	14
3	null	“cliente”	654251202	2024-08-12	2024-08-14	24.02	-1	15
4	0	null	654251202	2024-08-12	2024-08-14	24.02	-1	16

5	0	""	654251202	2024-08-12	2024-08-14	24.02	-1	17
6	0	"cliente"	0	2024-08-12	2024-08-14	24.02	-1	18
7	0	"cliente"	654251202	12/08/2024	2024-08-14	24.02	-1	19
8	0	"cliente"	654251202	2024-02-30	2024-08-14	24.02	-1	20
9	0	"cliente"	654251202	null	2024-08-14	24.02	-1	21
10	0	"cliente"	654251202	2024-08-12	14/02/2024	24.02	-1	22
11	0	"cliente"	654251202	2024-08-12	2024-02-30	24.02	-1	23
12	0	"cliente"	654251202	2024-08-12	null	24.02	-1	24
13	0	"cliente"	654251202	2024-08-12	2024-08-14	null	-1	25
14	0	"cliente"	654251202	2024-08-12	2024-08-14	-12.58	-1	26

### Casos de prueba para el método update de la clase ReservaRepository

Caso	id	cliente	movil	fechEnt	fechSal	precioTotal	Resultado esperado	Clases Cubiertas
Casos de prueba para las clases de equivalencia válidas								
1	0	"cliente"	654251202	2024-08-12	2024-08-14	24.02	> 1	1,2,3,4,5,6,7,8,9,10,11,12,13
Casos de prueba para las clases de equivalencia no válidas								
2	-1	"cliente"	654251202	2024-08-12	2024-08-14	24.02	-1	14
3	null	"cliente"	654251202	2024-08-12	2024-08-14	24.02	-1	15
4	0	null	654251202	2024-08-12	2024-08-14	24.02	-1	16
5	0	""	654251202	2024-08-12	2024-08-14	24.02	-1	17
6	0	"cliente"	0	2024-08-12	2024-08-14	24.02	-1	18
7	0	"cliente"	654251202	12/08/2024	2024-08-14	24.02	-1	19
8	0	"cliente"	654251202	2024-02-30	2024-08-14	24.02	-1	20
9	0	"cliente"	654251202	null	2024-08-14	24.02	-1	21
10	0	"cliente"	654251202	2024-08-12	14/02/2024	24.02	-1	22
11	0	"cliente"	654251202	2024-08-12	2024-02-30	24.02	-1	23
12	0	"cliente"	654251202	2024-08-12	null	24.02	-1	24
13	0	"cliente"	654251202	2024-08-12	2024-08-14	null	-1	25
14	0	"cliente"	654251202	2024-08-12	2024-08-14	-12.58	-1	26

## Casos de prueba para el método delete de la clase ReservaRepository

Caso	id	cliente	movil	fechEnt	fechSal	precioTotal	Resultado esperado	Clases Cubiertas
<b>Casos de prueba para las clases de equivalencia válidas</b>								
1	0	“cliente”	654251202	2024-08-12	2024-08-1 4	24.02	> 1	1,2,3,4,5,6,7,8,9 ,10,11,12,13
<b>Casos de prueba para las clases de equivalencia no válidas</b>								
2	-1	“cliente”	654251202	2024-08-12	2024-08-1 4	24.02	-1	15

Estos casos de prueba algunos generan fallos correctamente, pero muchas de estas pruebas se añaden a la base de datos sin generar ningún tipo de problemas, ya que en los inserts, update y del delete no es donde comprobamos que estos valores, no sean nulos o cumplan las determinadas condiciones, pero el usuario a la hora de añadir parcelas o reservas no puede incluir datos que sean nulos o fechas con formatos inválidos o fechas que pertenecen al calendario gregoriano.

Pero si que al ejecutar las pruebas e intentar añadir un número de teléfono como null, da error y no se puede insertar.

## Pruebas de volumen

Para evaluar la capacidad de manejo de grandes volúmenes de datos en la aplicación, es fundamental realizar pruebas de volumen. Estas pruebas simulan la inserción masiva de datos, como parcelas y reservas, con el fin de identificar posibles problemas relacionados con el rendimiento y la estabilidad del sistema cuando se trabaja con grandes cantidades de información.

En este caso, se han diseñado bucles que insertan 100 parcelas y 10,000 reservas. Una vez ejecutadas las pruebas, se ha verificado que todas las parcelas y reservas se han añadido correctamente, ya que no se ha producido ningún fallo en el sistema y el programa sigue funcionando de manera estable y eficiente.

## Pruebas de sobrecarga

Para abordar las pruebas de sobrecarga con descripciones de parcelas, se ha desarrollado un programa que incrementa gradualmente el tamaño de las descripciones de 10,000 en 10,000 caracteres. El objetivo de esta prueba es evaluar cómo maneja el sistema descripciones extremadamente largas, verificando su capacidad antes de que surjan problemas como la no respuesta o errores relacionados con la memoria.

Los resultados muestran que hasta los 210,000 caracteres, la aplicación puede continuar funcionando, aunque el dispositivo muestra advertencias de que la aplicación no responde. Al seleccionar la opción de esperar, el programa continúa ejecutándose y añadiendo descripciones con valores superiores. Sin embargo, es evidente una disminución en el rendimiento, ya que las parcelas con descripciones extremadamente largas tardan más en cargarse para acceder a sus datos.

Además, al intentar acceder o editar parcelas con descripciones extremadamente largas, se observó que el límite máximo aceptable para un TextView es de aproximadamente 20,000 caracteres.