**Proyecto IT – 2024/2025**

**Aldapa**



Daniel Sánchez-Matamoros Carmona

Pablo Falcón Rodríguez

# Resumen ejecutivo

Enlace al repositorio de GitHub:<https://github.com/Pablichi/Trabajo-IT.git>

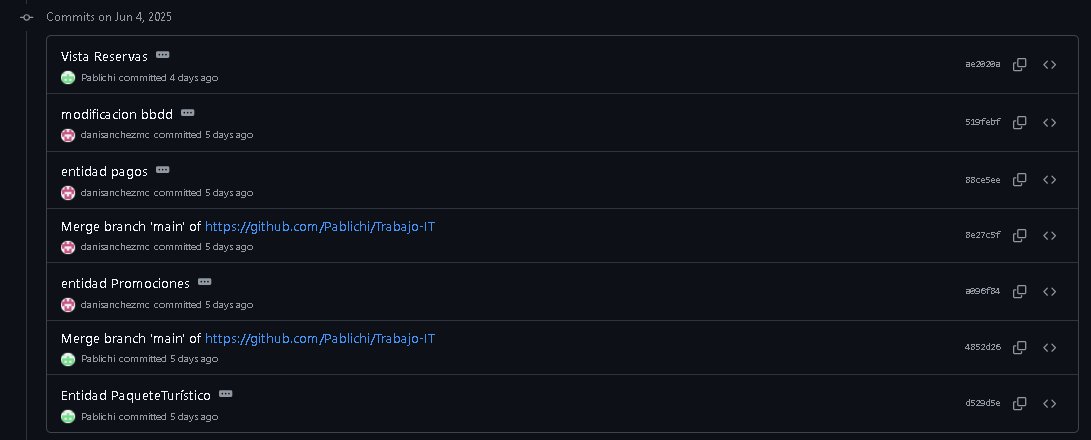
A continuación, se muestran capturas con las contribuciones y commits (ordenados por fecha de más antiguo a más nuevo) de cada uno de los integrantes del grupo.

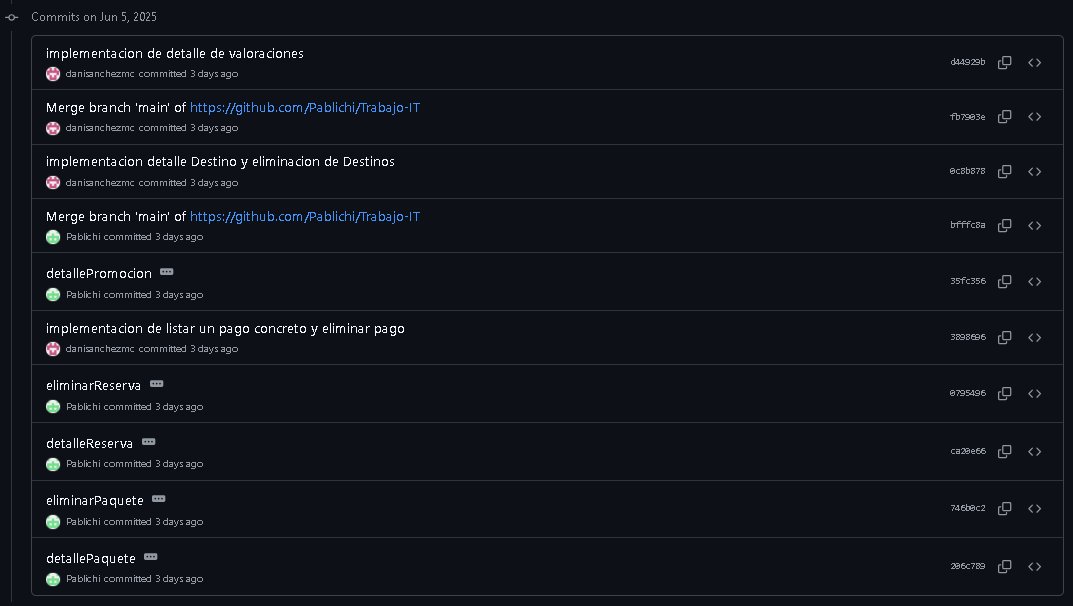
Pantalla de vídeo juego

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

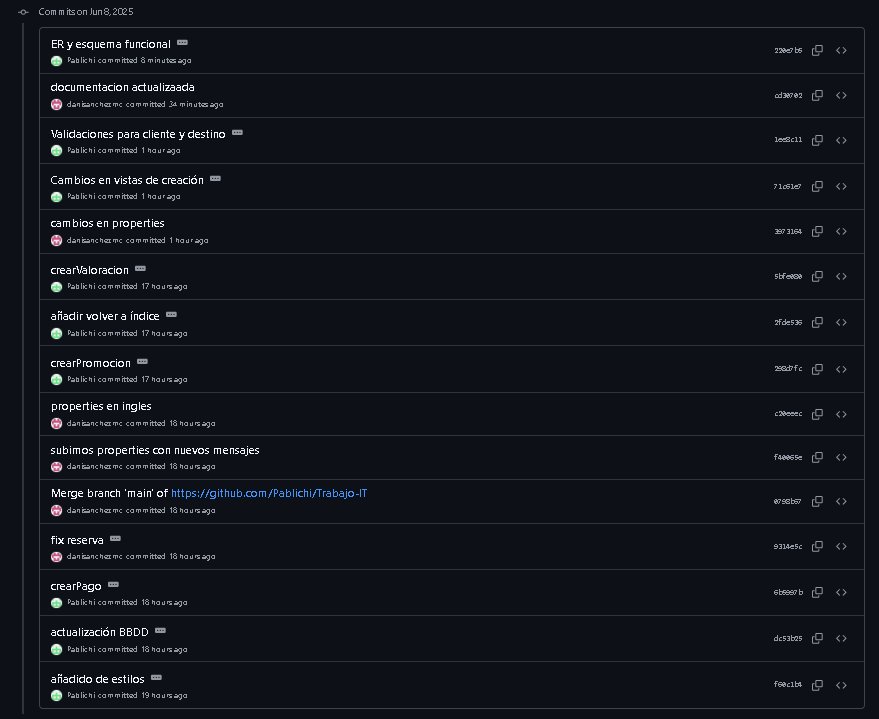


Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

# Entidades tratadas y operaciones CRUD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Create** | **Read** | **Update** | **Delete** |
| Cliente |  |  |  |  |
| Paquete Turístico |  |  |  |  |
| Promoción |  |  |  |  |
| Destino |  |  |  |  |
| Pago |  |  |  |  |
| Reserva |  |  |  |  |
| Valoración |  |  |  |  |
| Proveedor de Servicios |  |  |  |  |

# Aportaciones extraordinarias

Hemos implementado validaciones programáticas y validaciones con anotaciones en las creaciones de las entidades para los diferentes tipos de campos, haciendo algunos campos obligatorios, cumpliendo requisitos en otros campos, para dar mejor calidad al código y una mejor experiencia de usuario, aunque no se pidiera en el proyecto.

Además, hemos implementado la internalización de los mensajes de la aplicación para tenerla disponible tanto en español como inglés.

En la implementación de los CRUDs, no nos hemos limitado sólo a que funcionen las operaciones, sino que le hemos dado un poco más de complejidad al combinar tablas y ofrecer la posibilidad al usuario en las ventanas de creación, de listar valores en campos tipo option, donde estos a su vez hacen internamente otro listar de otra entidad, y asi los campos del formulario no son solo inputText, sino que combinamos select, radio, fechas... con el objetivo de que el proyecto sea lo más cercano a la realidad posible.

Para generar un mayor atractivo visual, hemos implementado hojas de estilo CSS para cada vista.

Aunque se pedían 2 entidades por componente, hemos implementado el doble cada uno, debido a que hemos tenido que hacer el proyecto entre 2 componentes en lugar de 4.

# Problema a resolver

Actualmente, muchas personas interesadas en viajar se enfrentan a la dificultad de buscar, comparar y reservar experiencias turísticas a través de múltiples medios, lo que fragmenta la información y complica la planificación. A esto se suma la falta de seguimiento eficiente tras la reserva, así como la escasa personalización en la oferta de paquetes. En respuesta a esta problemática, este proyecto propone el desarrollo de una plataforma web integral que centralice y optimice la gestión de viajes turísticos.

El objetivo principal de la plataforma es simplificar la experiencia del cliente, permitiéndole explorar, reservar y dar seguimiento a paquetes turísticos desde una sola interfaz, intuitiva y accesible. Desde la página de inicio, los usuarios pueden navegar por un catálogo de destinos organizados según tipo de experiencia, localización geográfica y temporada ideal. Cada destino cuenta con una vista detallada que incluye los paquetes disponibles, itinerarios, precios, duración, fechas de salida, servicios incluidos y posibles promociones aplicables, lo que facilita la toma de decisiones.

Además de mejorar la experiencia del cliente, la plataforma también busca resolver las necesidades de gestión interna de una agencia de viajes. Para ello, los usuarios pueden acceder a un panel con módulos de gestión que permiten realizar operaciones CRUD sobre entidades clave como destinos, paquetes turísticos, clientes, reservas, pagos, cuentas de usuario, proveedores de servicios, promociones y valoraciones.

Esta funcionalidad administrativa integral permite mantener actualizada la oferta, llevar un control financiero de las transacciones y asegurar la calidad de los servicios ofrecidos.

Por último, las promociones ofrecen herramientas de marketing efectivas, permitiendo aplicar descuentos o beneficios especiales a determinados paquetes. Las valoraciones de los clientes, por su parte, fortalecen la transparencia del sistema y promueven la mejora continua, al reflejar la experiencia real de quienes han contratado los servicios.

Aunque inicialmente habíamos previsto implementar 10 entidades, finalmente quedará en 8 dejando fuera de implementación las entidades administrador y cuenta, serán aspectos que no cubriremos, aunque sí que dejaremos la entidad en base de datos creada con sus campos, el servicio REST implementado y documentado, pero no implementaremos CRUD para estas entidades.

# Recolección de los requerimientos del problema

Navegación y exploración de destinos**:**  
Los usuarios podrán visualizar un catálogo de destinos turísticos clasificados por tipo, región y temporada recomendada, incluyendo descripciones.

Visualización de paquetes turísticos:  
Los usuarios podrán consultar paquetes turísticos asociados a un destino específico, con detalles como itinerario, precio, duración, cupo, servicios incluidos, y promociones activas.

Gestión de reservas**:**  
El sistema permitirá a los clientes realizar reservas de paquetes turísticos y seleccionando promociones en caso de que haya alguna vigente.

Procesamiento de pagos**:**  
Los clientes podrán registrar pagos asociados a sus reservas mediante diferentes medios (tarjeta, transferencia, etc.). El sistema registrará la transacción y su estado (pendiente, aprobado, rechazado).

Gestión de promociones**:**  
Los administradores podrán crear, modificar, eliminar y asignar promociones a paquetes turísticos. Cada promoción podrá tener condiciones específicas (porcentaje de descuento, fechas de validez, etc.).

Valoraciones de paquetes turísticos**:**  
Los clientes pueden dejar una valoración sobre el paquete adquirido, incluyendo puntuación y comentarios. Estas valoraciones serán visibles para otros usuarios.

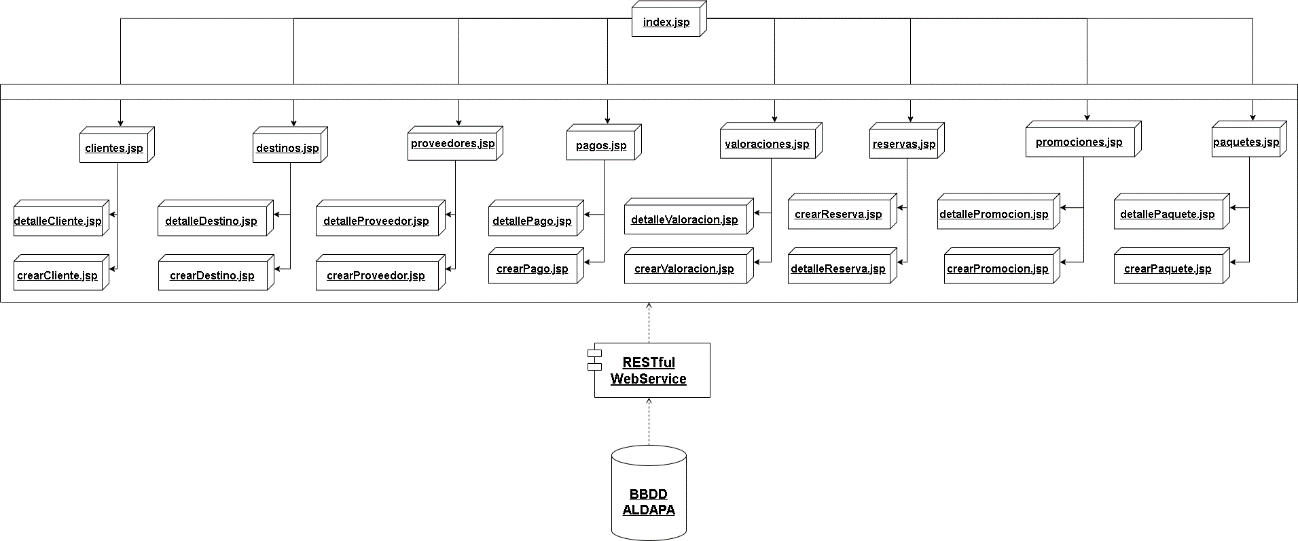
Gestión de valoraciones**:**  
Las valoraciones podrán ser revisadas, mostrándose en un listado, pudiendo ser moderadas con la posibilidad de eliminar aquellas que incumplan las políticas del sistema.

Gestión administrativa del sistema**:**  
Los usuarios tendrán acceso a un panel desde el cual podrán realizar operaciones CRUD sobre todas las entidades del sistema: destinos, paquetes turísticos, clientes, reservas, pagos, proveedores de servicios, promociones y valoraciones.

Historial de reservas de clientes**:**  
Se podrá acceder a un listado de reservas, ver detalles, estado de pago y crear nuevas reservas si se desea.

# Esquema funcional

En el siguiente esquema podemos visualizar el flujo normal de la aplicación:



Contamos con un servicio web RESTful que es el encargado de acceder a la base de datos. Todas las acciones que requieren operaciones CRUD en las entidades del sistema utilizan este servicio como intermediario.

En primer lugar, contamos con un action llamado cargaInicial, donde iniciamos en sesion todas las listas de todas las entidades, con el fin de disponer de todas ellas cuando sea necesario.

Contamos con una página principal llamada index.jsp, desde esta página, los usuarios pueden seleccionar una de las 8 opciones disponibles, cada una representa un listado de las entidades de la aplicación detalladas anteriormente.

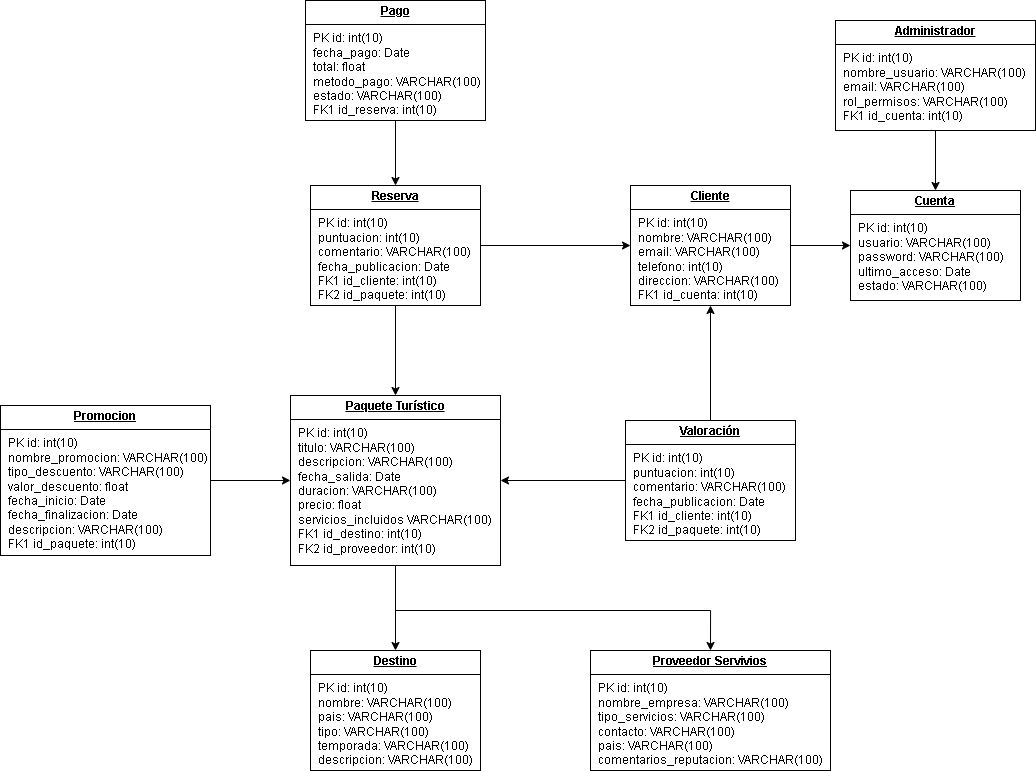
Una vez seleccionada una opción, además de poder ver el listado correspondiente a esa entidad, tendremos un botón con la funcionalidad de crear un objeto de esa entidad o pulsar sobre un elemento del listado y pasar a la vista del detalle de esa entidad seleccionada.

Desde la vista detalle, tenemos la opción de eliminar la entidad mostrada, volver al listado anterior o volver a la pantalla inicial.

**BBDD aldapa:** Almacena toda la información de las entidades del sistema.

# Diseño de la base de datos

Para el diseño de la base de datos hemos elaborado un diagrama entidad-relación, así como una justificación de las entidades y relaciones que lo contiene.



1. **Destino**
   * Almacena características geográficas y turísticas.
2. **Paquete turístico**
   * Relaciona destinos y proveedores para construir paquetes.
   * float para precio permite decimales
3. **Cliente**
   * Cada cliente puede estar vinculado a una cuenta.
4. **Reserva**
   * Conecta clientes con paquetes. estado permite controlar proceso.
5. **Administrador**
   * Asociación con la cuenta del sistema para controlar accesos.
   * Clave foránea con ON DELETE SET NULL permite preservar registros.
6. **Cuenta**
   * Tabla central de autenticación. Es referenciada por cliente y administrador.
   * Tipos de datos apropiados por longitud y seguridad.
7. **Pago**
   * Registra transacciones. estado útil para control (pagado, pendiente, etc.).
8. **Proveedor de servicios**
   * Detalla las empresas que brindan los servicios de los paquetes.
9. **Promoción**
   * Permite aplicar descuentos según campaña o fechas.
10. **Valoración**
    * Permite que los clientes evalúen paquetes.

## Relaciones entre Tablas - Esquema Resumido

### **🧾 cuenta**

* Relacionada con:
  + cliente → id\_cuenta (FK)
  + administrador → id\_cuenta (FK)

### **👤 cliente**

* Relacionada con:
  + cuenta (1 cuenta puede tener muchos clientes)
  + reserva → id\_cliente (FK)
  + valoracion → id\_cliente (FK)

### **👨‍💼 administrador**

* Relacionada con:
  + cuenta → id (FK)

### **📍 destino**

* Relacionada con:
  + paquete\_turistico → id\_destino (FK)

### **🏢 proveedor\_servicios**

* Relacionada con:
  + paquete\_turistico → id\_proveedor (FK)

### **📦 paquete\_turistico**

* Relacionada con:
  + destino (lugar del paquete)
  + proveedor\_servicios (quién presta los servicios)
  + reserva → id\_paquete (FK)
  + promocion → id\_paquete (FK)
  + valoracion → id\_paquete (FK)

### **📅 reserva**

* Relacionada con:
  + cliente (quién reserva)
  + paquete\_turistico (qué paquete reserva)
  + pago → id\_reserva (FK)

### **💸 pago**

* Relacionada con:
  + reserva (por cuál reserva se pagó)

### **🎁 promocion**

* Relacionada con:
  + paquete\_turistico (qué paquete está en promoción)

### **⭐ valoracion**

* Relacionada con:
  + cliente (es quien da la valoración)
  + paquete\_turistico (qué paquete valora)

## Resumen de relaciones:

* Un **cliente** tiene una **cuenta**, puede hacer **reservas**, pagar por ellas y dejar **valoraciones** de paquetes.
* Un **administrador** también tiene una **cuenta**, con permisos específicos.
* Una **reserva** está hecha por un **cliente** para un **paquete turístico**, y se relaciona con un **pago**.
* Un **paquete turístico** pertenece a un **destino**, es ofrecido por un **proveedor** y puede tener **promociones** y **valoraciones**.
* Un **pago** está vinculado a una única **reserva**.

# Diseño físico de la base

🔐 **1. Tabla: cuenta**

* **PK:** id
* **Atributos:**
  + usuario (varchar(100)): Nombre de inicio de sesión.
  + password (varchar(100)): Contraseña del usuario.
  + ultimo\_acceso (date): Última fecha en que accedió.
  + estado (varchar(100)): Activo/Inactivo, etc.

### **👨‍💼 2. Tabla: administrador**

* **PK:** id
* **FK:** id\_cuenta → cuenta(id)
* **Atributos:**
  + nombre\_usuario (varchar(100)): Nombre del admin.
  + email (varchar(100)): Email de contacto.
  + rol\_permisos (varchar(100)): Permisos asignados (ej. CRUD).

### **👤 3. Tabla: cliente**

* **PK:** id
* **FK:** id\_cuenta → cuenta(id)
* **Atributos:**
  + nombre, email, direccion (varchar(100)): Información básica.
  + telefono (int): Puede considerarse varchar si incluye signos o prefijos.

### **📍 4. Tabla: destino**

* **PK:** id
* **Atributos:**
  + nombre, pais, tipo, temporada, descripcion (varchar): Detalles del destino.

### **📦 5. Tabla: paquete\_turistico**

* **PK:** id
* **FKs:**
  + id\_destino → destino(id)
  + id\_proveedor → proveedor\_servicios(id)
* **Atributos:**
  + título, descripción, servicios\_incluidos (varchar): Descripción general.
  + fecha\_salida (date), duracion (int), precio (float)

### **💰 6. Tabla: pago**

* **PK:** id
* **FK:** id\_reserva → reserva(id)
* **Atributos:**
  + fecha\_pago (date), total (float), metodo\_pago (varchar), estado (varchar)

### **📑 7. Tabla: reserva**

* **PK:** id
* **FKs:**
  + id\_cliente → cliente(id)
  + id\_paquete → paquete\_turistico(id)
* **Atributos:**
  + fecha\_reserva (date), numero\_personas (int), estado (varchar), precio\_total (float)

### **🏷️ 8. Tabla: promocion**

* **PK:** id
* **FK:** id\_paquete → paquete\_turistico(id)
* **Atributos:**
  + nombre\_promocion, tipo\_descuento, descripcion (varchar)
  + valor\_descuento (float), fecha\_inicio, fecha\_finalizacion (date)

### **🤝 9. Tabla: proveedor\_servicios**

* **PK:** id
* **Atributos:**
  + nombre\_empresa, tipo\_servicios, contacto, pais, comentarios\_reputacion (varchar)

### **⭐ 10. Tabla: valoracion**

* **PK:** id
* **FKs:**
  + id\_cliente → cliente(id)
  + id\_paquete → paquete\_turistico(id)
* **Atributos:**
  + puntuacion (int), comentario (varchar), fecha\_publicacion (date)

# Desarrollo de casos de prueba

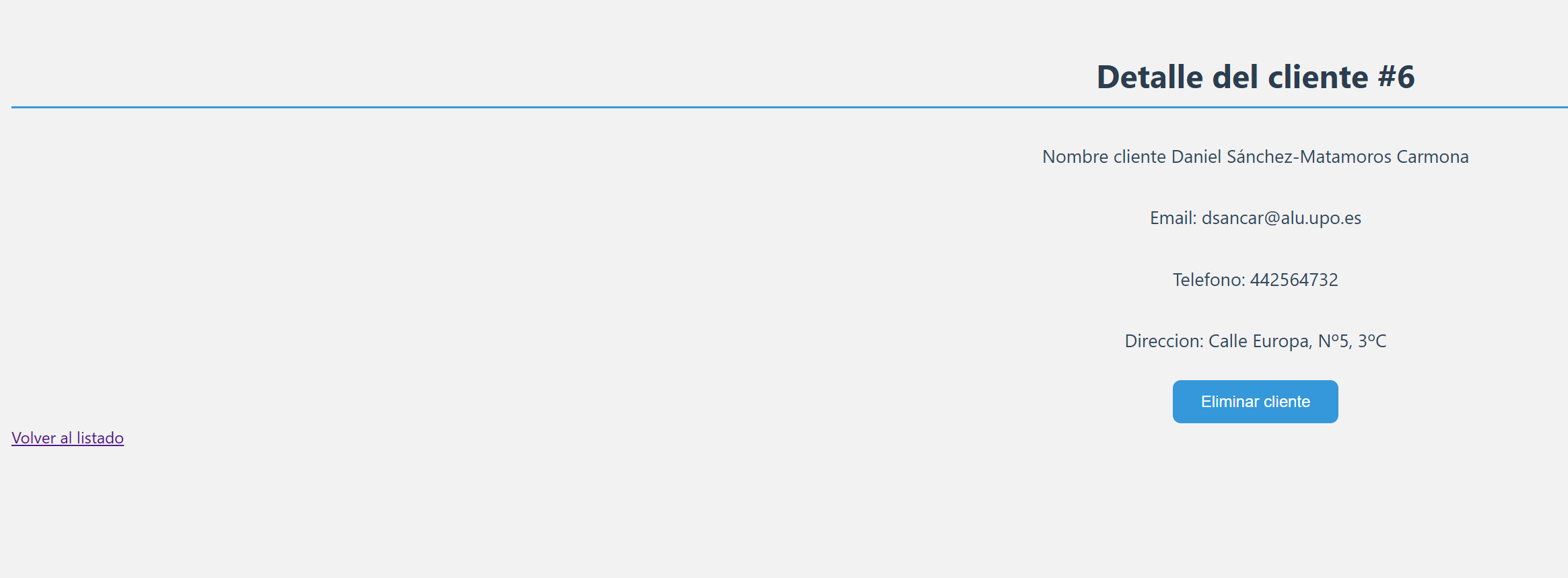
En este apartado vamos a documentar una batería de pruebas que hemos diseñado probando diferentes acciones que podemos realizar y sobre diferentes entidades:

## Consulta de un cliente específico

Primero listamos todos los clientes:

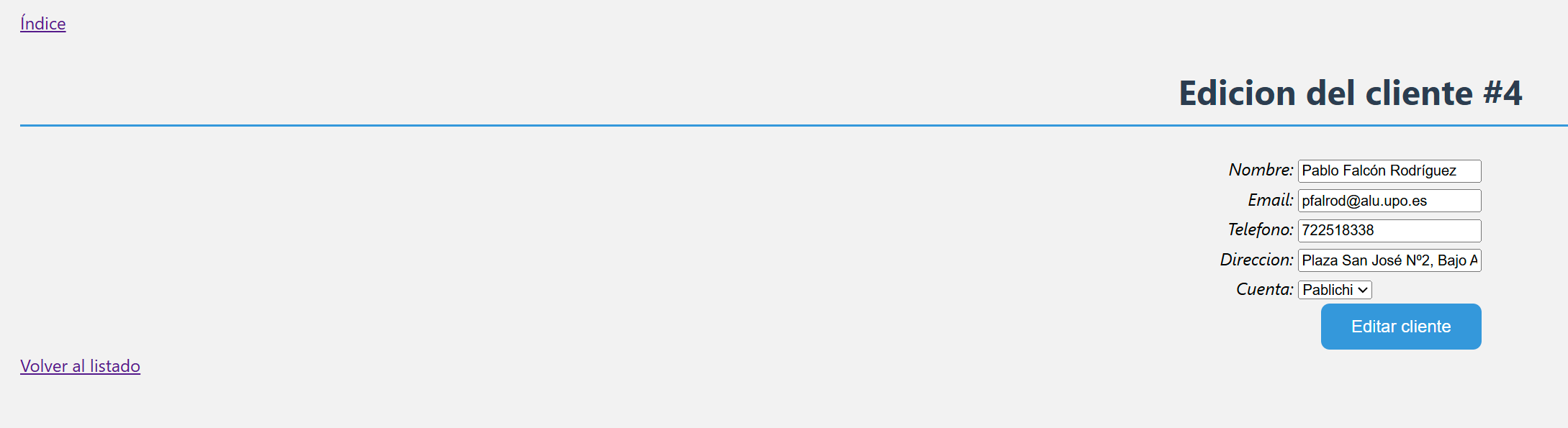


posteriormente hacemos click en uno de ellos:



## Editar un cliente especifico

Del listado anterior, seleccionamos “editar cliente” sobre uno en concreto:

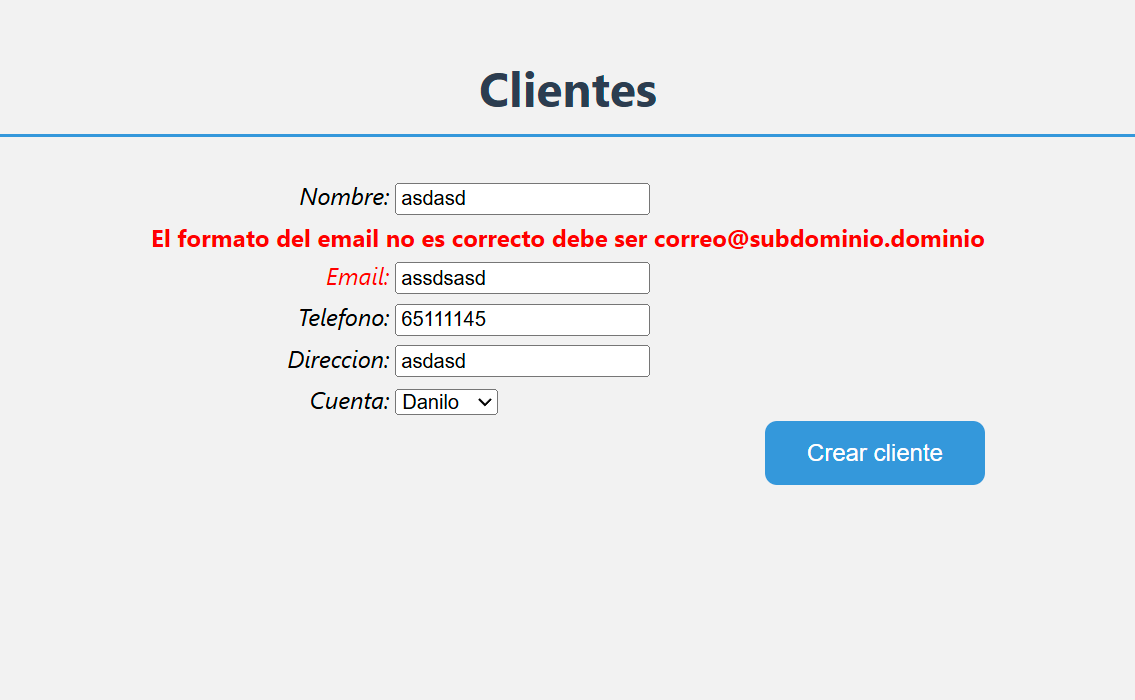


Cambiamos los datos que queramos y pulsamos sobre editar cliente, si todo ha ido bien nos redirigirá al listado de clientes de nuevo

Si deseamos volver atrás, podemos hacerlo con los enlaces de la izquierda, podemos volver tanto al listado de clientes como al menú principal.

## Creación de un cliente

Desde la vista de listado, pulsamos sobre el botón crear nuevo cliente y nos aparecerá un formulario donde debemos rellenar los datos de cliente



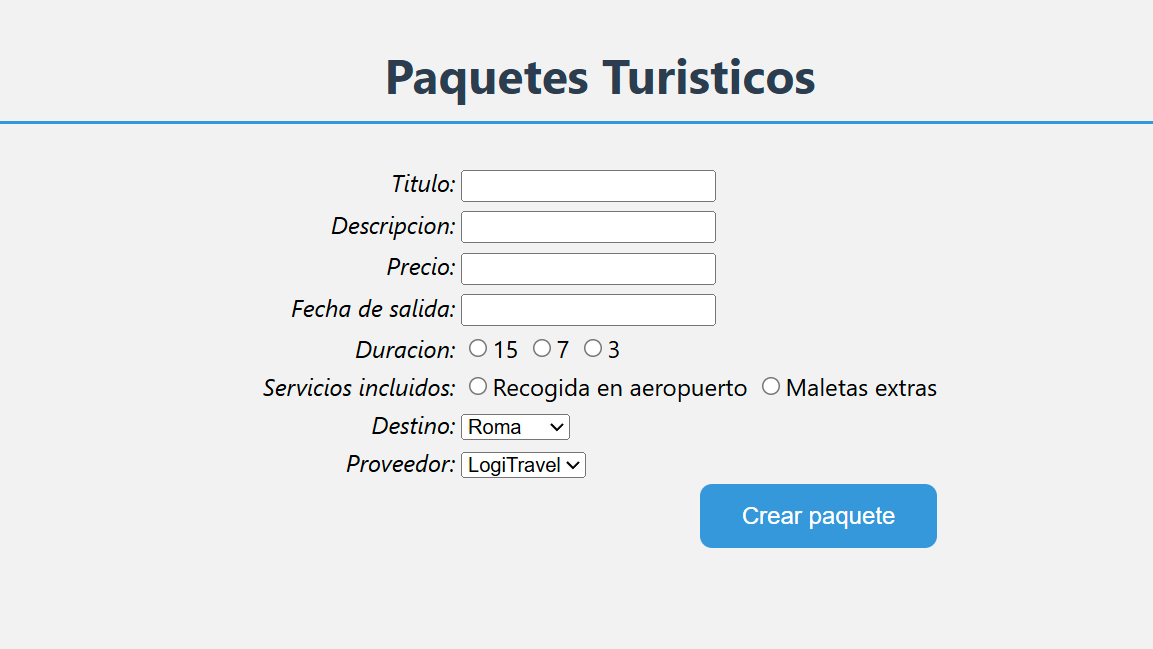
Si no rellenamos los datos correctamente la aplicación nos avisará cuando valide los campos al pulsar el botón crear cliente

## Crear un paquete turístico

Primero accedemos al listado de paquetes

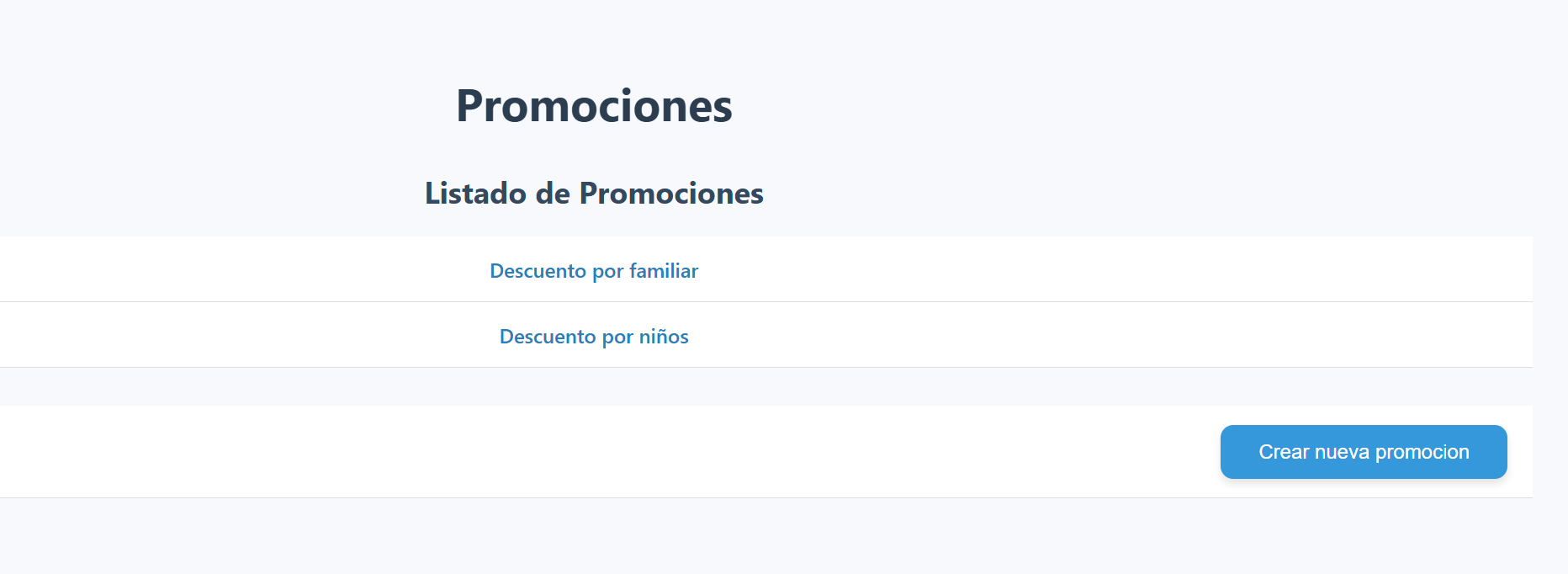


Pulsamos sobre crear, y accedemos al formulario de creación de paquetes donde además de elegir los campos relacionados con el paquete, podemos elegir un destino desde el listado de destinos que se despliegan en el combo, como el proveedor mostrando los proveedores disponibles, una vez creado satisfactoriamente, volveríamos al listado de paquetes de nuevo.



## Ver detalle de una promoción en concreto

Listamos las promociones existentes



Hacemos clic en una de ellas:



Podemos ver sobre que paquete aplicaría la promoción, las fechas en las que esta activa, el valor del descuento y otros datos como la descripción.

Si deseamos eliminar la promoción solo tendríamos que pulsar sobre eliminar promoción

# Manual de uso

En este apartado vamos a detallar los pasos a seguir para poder hacer uso de la aplicación.

En primer lugar, necesitamos desplegar la base de datos, para ello debemos contar con la herramienta XAMPP, y arrancar los servicios de Apache para tener un servidor web y MySQL para tener disponible un servidor de base de datos.

Una vez hecho esto, entramos en la página web mySQLAdmin del servidor de base de datos que hemos desplegado anteriormente, desde ahí tenemos que importar la base de datos Aldapa.sql que facilitamos en la entrega de los materiales del proyecto.

Una vez hecha la importación, observamos que ya contamos con las tablas, atributos y datos en nuestra base de datos, pasamos ahora a desplegar el proyecto.

Desde el IDE Netbeans, abrimos los dos proyectos facilitados en la entrega. Por una parte, el proyecto RESTaldapa, el cuál debemos construir y ejecutar primero, previamente debemos disponer del servicio GlassFish server disponible, para comprobarlo lo hacemos desde la pestaña prestaciones y Servidores. Comprobamos que cuando arrancamos el proyecto, se debe desplegar en este apartado el servicio REST.

Una vez desplegado el servicio REST, debemos construir y ejecutar el proyecto EBaldapa, el cual, abrirá automáticamente una ventana web con la ventana index mostrando el menú principal con las diferentes funcionalidades de la aplicación y estará listo para usarse.