

PRÁCTICA 2 : Reingeniería software con *código heredado*

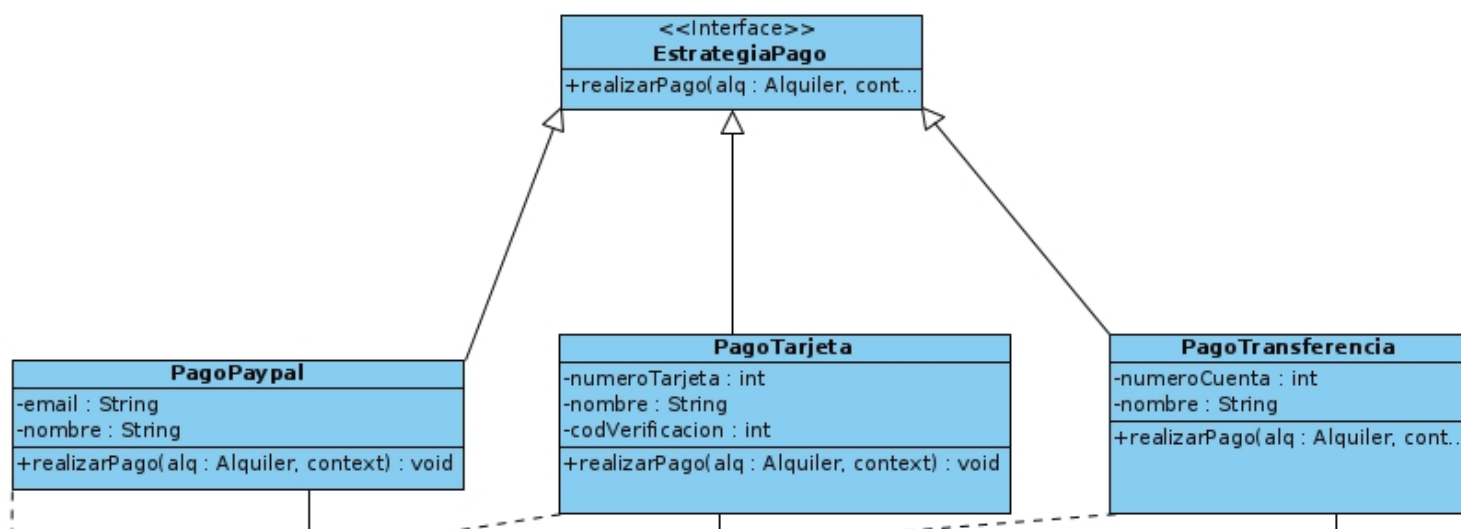
Cristian Fernández Jiménez
Ángel Gómez Ferrer

El objeto de trabajo se trata de una aplicación para encuentros deportivos.

Concepción de la práctica y reutilización de implementación pasada

Como aplicación a desarrollar hemos consensuado la realización del alquiler completo de una pista, con un deporte seleccionado y una fecha concreta. Además, se consultará disponibilidad para realizarlo, almacenándolo si es posible y llevándonos a una pasarela de pago, según el método que elijamos. Todo ello mediante Android Studio y una aplicación totalmente operativa.

Añadiremos por lo tanto un nuevo patrón de diseño: Estrategia. Con él encapsularemos la lógica de pagar un precio por un alquiler, pudiendo añadir varios en un futuro para que cuente la aplicación.



Además, reutilizaremos y aplicaremos los patrones de la anterior práctica: Filtros. En nuestro caso, nos permitía modificar el alquiler que estamos realizando, según la pista y los jugadores, consiguiendo el precio a pagar finalmente por el usuario que alquile la ubicación.

Creación de aplicación Móvil

Como aplicación crearemos una ventana inicial que contenga la posibilidad de Alquilar una ubicación. Pulsando este botón nos redirigirá a la creación del alquiler, que será totalmente personalizada y con desplegados que permitan seleccionar las ubicaciones, deportes y métodos de pago para realizar el alquiler. Finalmente, con los datos rellenos, pasaremos por la pasarela de pago, introduciremos los datos y se realizará el pago del precio calculado por los filtros.

Los **requisitos funcionales** nuevos son:

- Realizar un alquiler.
- Realizar el pago de este.
- Consultar disponibilidad de alquileres.
- Alertar de errores ó problemas de disponibilidad de pistas.

Requisitos no funcionales:

- Se desarrollará una APP programada en Flutter
- Será muy visual, intuitiva y con ayuda al usuario.

Refactorización

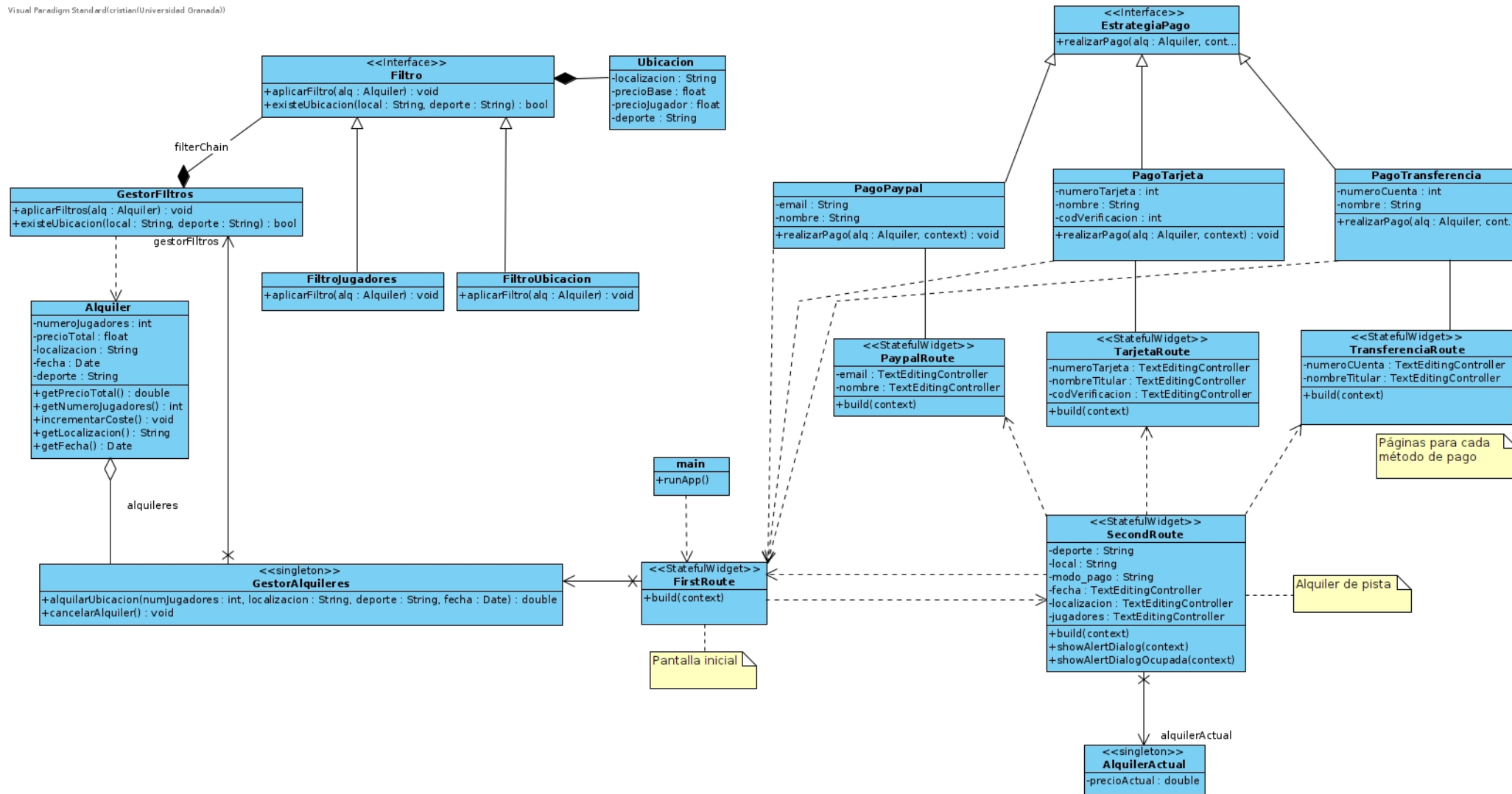
En la tarea de reingeniería y reutilización del código existente, hemos necesitado añadir un gestor de alquileres que permita la creación y almacenamiento de estos. Así, podrán existir varios alquileres simultáneamente que podrán probar la aplicación.

Este gestor entonces reutilizará y aplicará los filtros ya implementados, pues contará con un gestor de filtros como atributo, además de una lista de alquileres ya realizados y pagados.

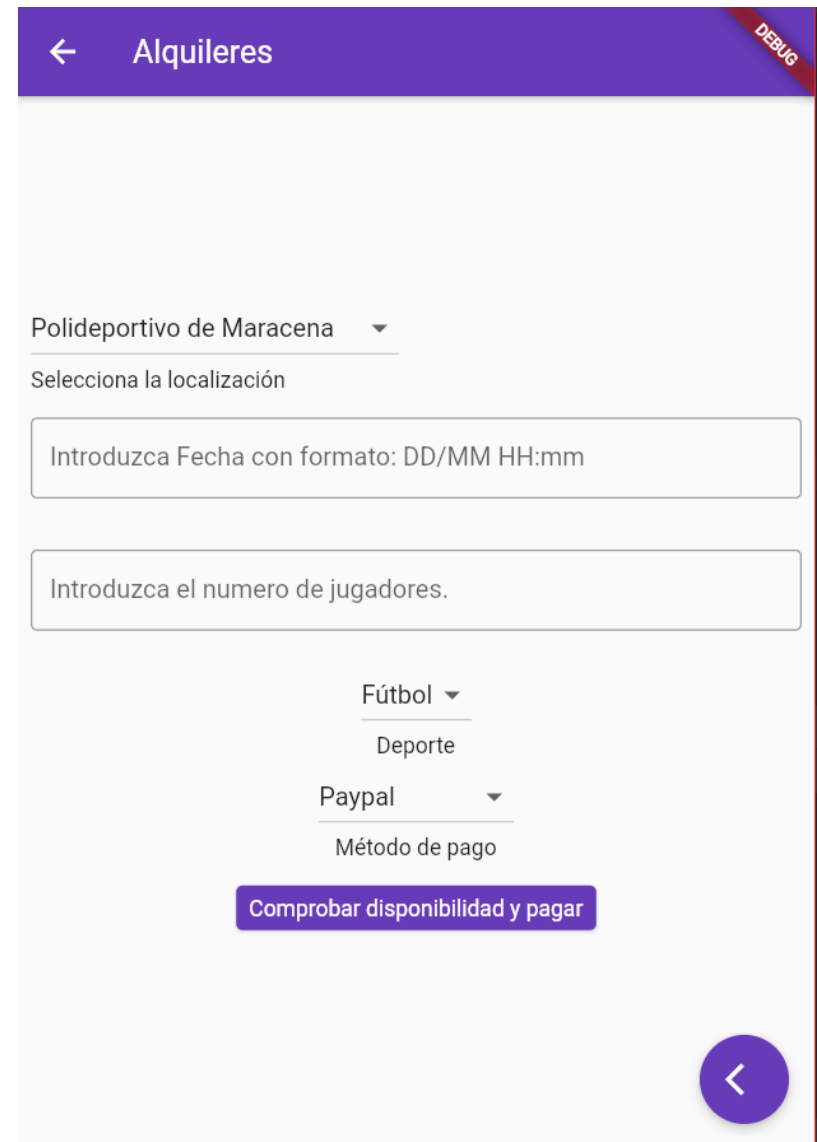
Por lo tanto, este gestor será el encargado de vincular la lógica de los alquileres con la aplicación a desarrollar.



Diagrama UML – CLASES

Visual Paradigm Standard(cristian(Universidad Granada))



Aplicación en Flutter



 Alquileres 

Polideportivo de Maracena ▼

Selecciona la localización


Fútbol ▼


Deporte

Paypal ▼

Método de pago


Comprobar disponibilidad y pagar

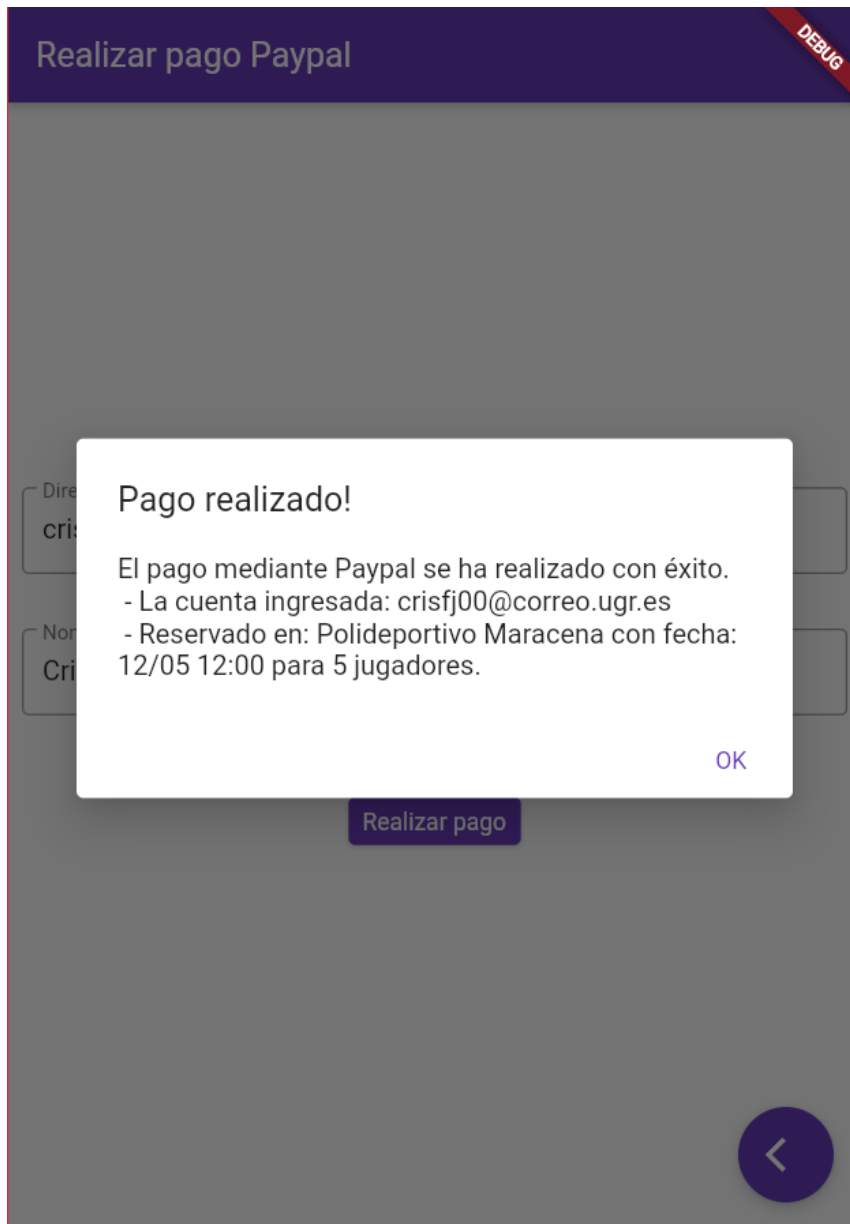


Realizar pago Paypal 

Total a pagar: 5.5€

Realizar pago





Comentar finalmente que las páginas de Pago Tarjeta y Pago Transferencia están también disponibles pero las hemos omitido en el documento.

La variación entre estas son los campos de entrada de texto y el mensaje final.