Iteración 3

Pedro L. Lobato Barros

Universidad de los Andes, Bogotá, Colombia

{p.lobato[}@uniandes.edu.co](mailto:%7d@uniandes.edu.co)

Fecha de presentación: Mayo 06 de 2023

Tabla de contenido

[1 Introducción 1](#_Toc134822388)

[2 Modelos 2](#_Toc134822389)

[2.1 Modelo de clases 2](#_Toc134822390)

[2.2 Modelo relacional 10](#_Toc134822391)

[3 Resultados 12](#_Toc134822392)

# Introducción

La aplicación que se desarrolló es una aplicación Java que utiliza el marco de trabajo *Java Data Objects* (*JDO*) para administrar operaciones transaccionales y de consulta en una base de datos relacional. *JDO* proporciona una capa de abstracción que permite al programador trabajar con objetos Java y dejar que el proveedor de *JDO* se encargue de la persistencia de los datos en la base de datos.

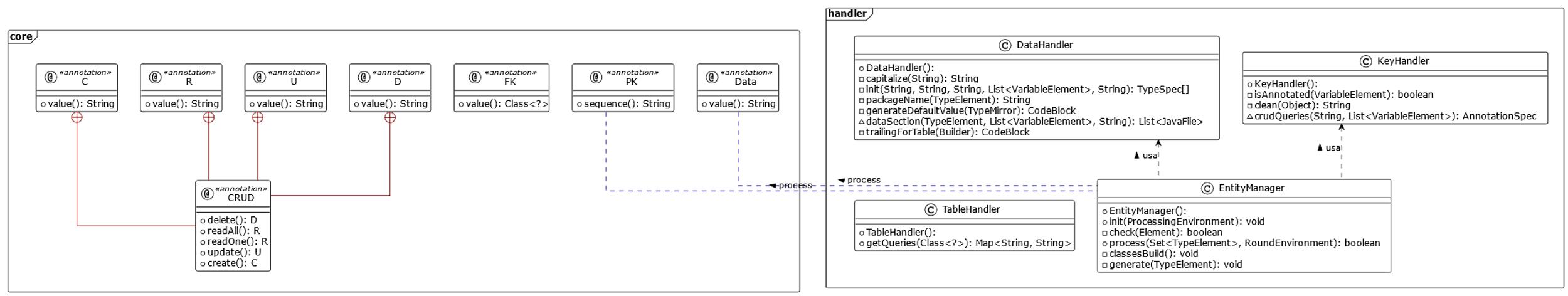
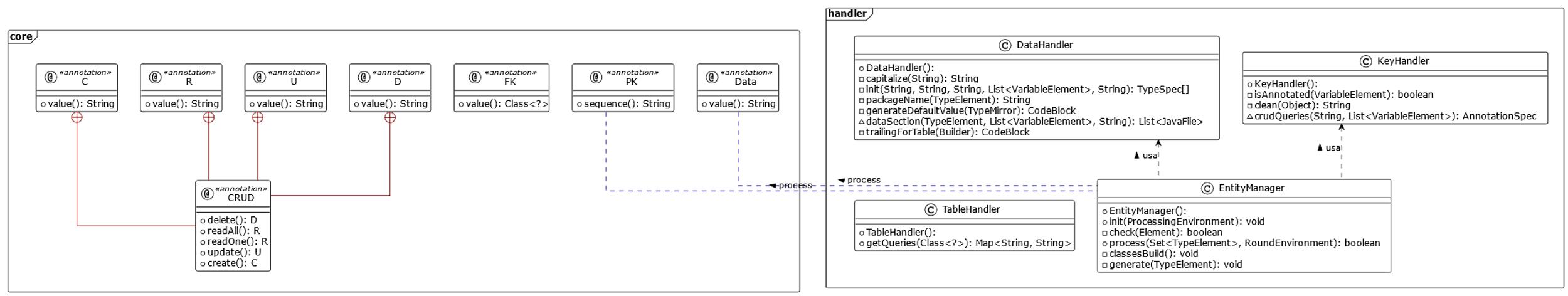
La aplicación utiliza operaciones *CRUD* (Crear, Leer, Actualizar y Eliminar) para interactuar con la base de datos, así como operaciones de consulta para responder a los requerimientos funcionales del usuario. Además, la aplicación utiliza código SQL embebido para realizar completar dichas operaciones más avanzadas en la base de datos.

# Modelos

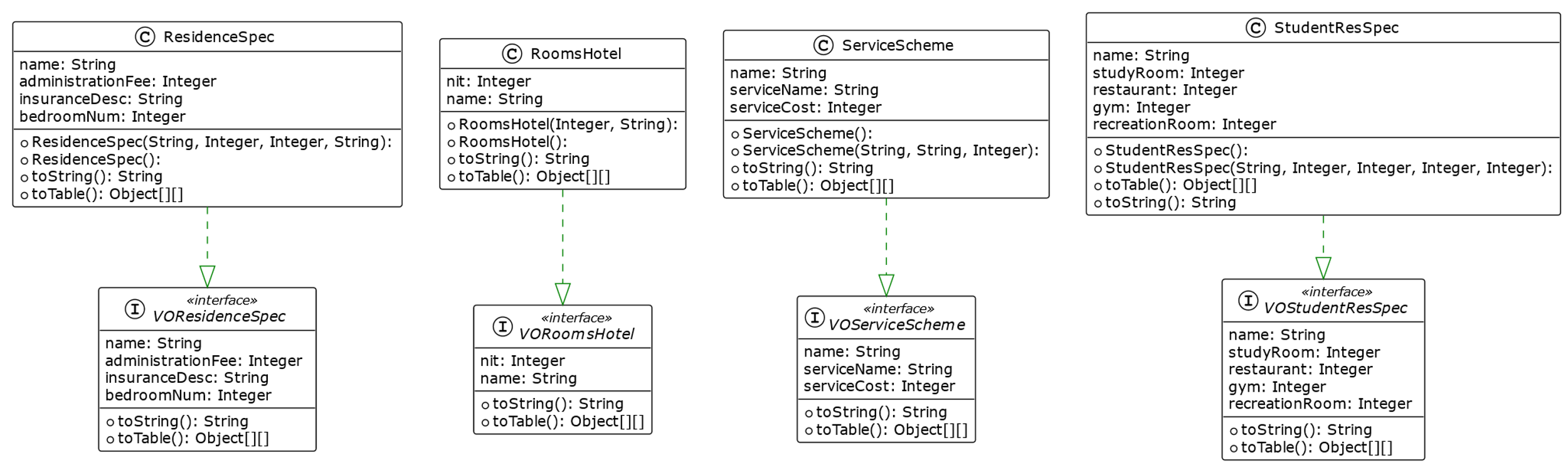
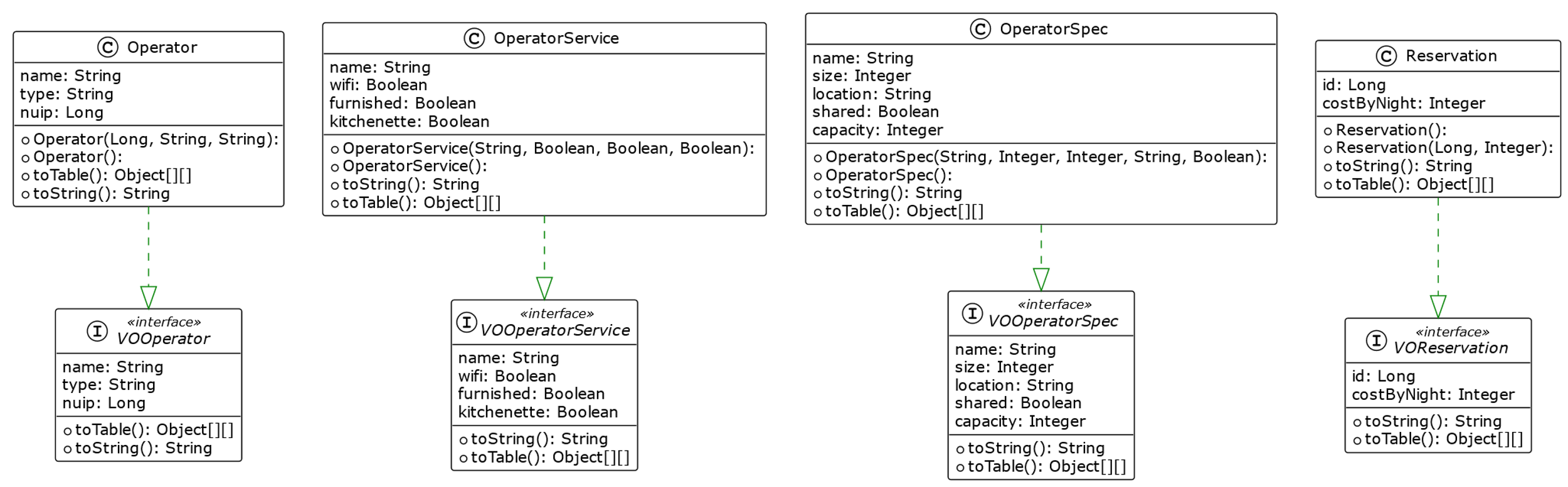
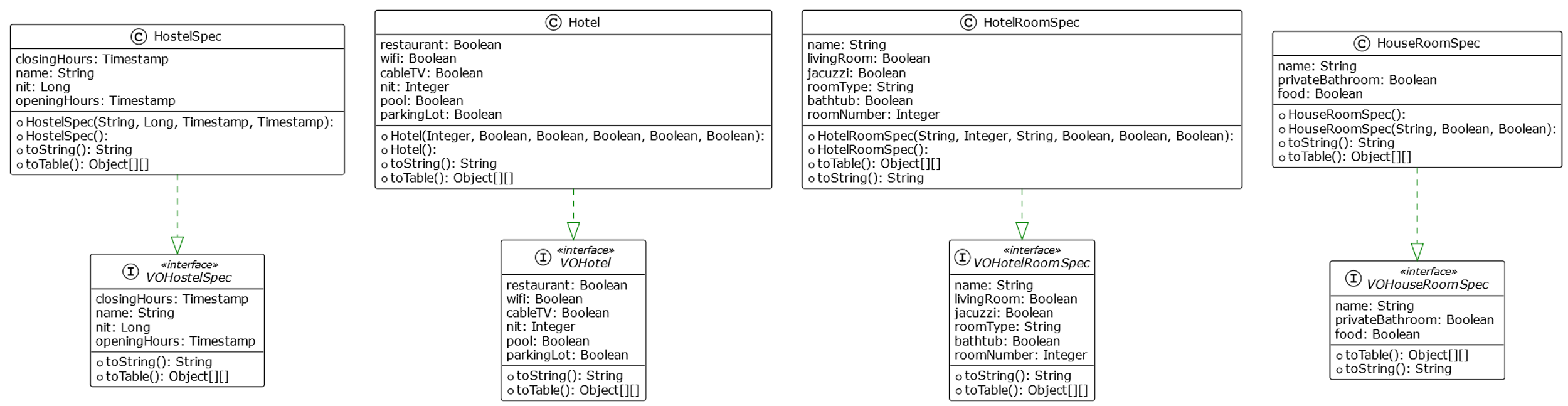
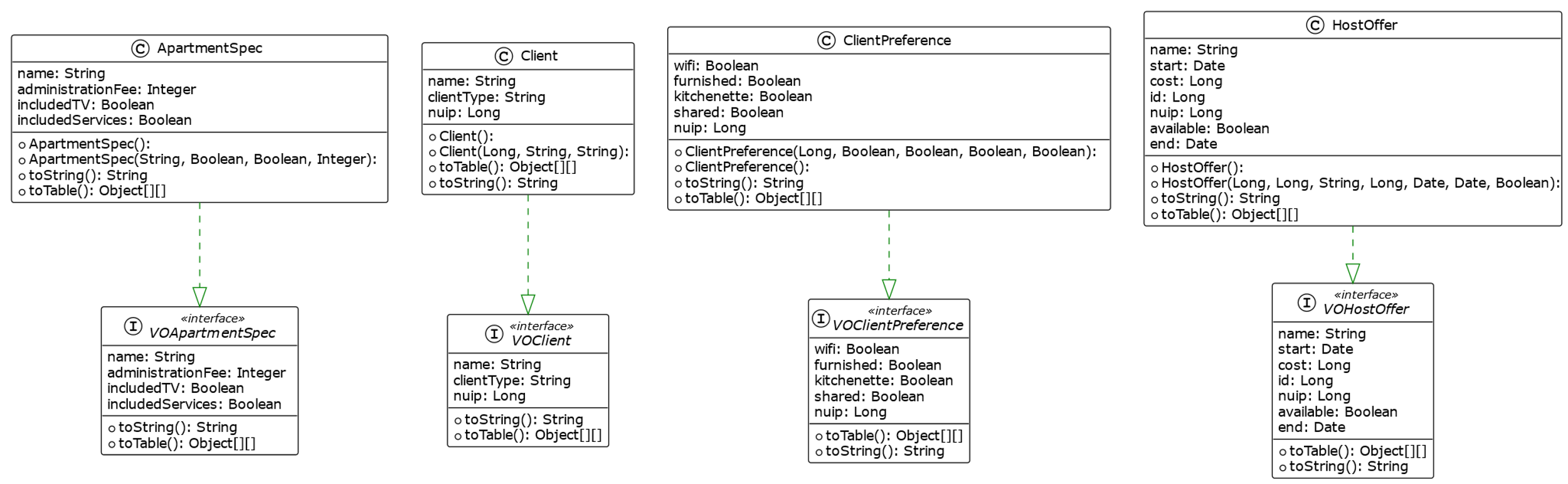
## Modelo de clases

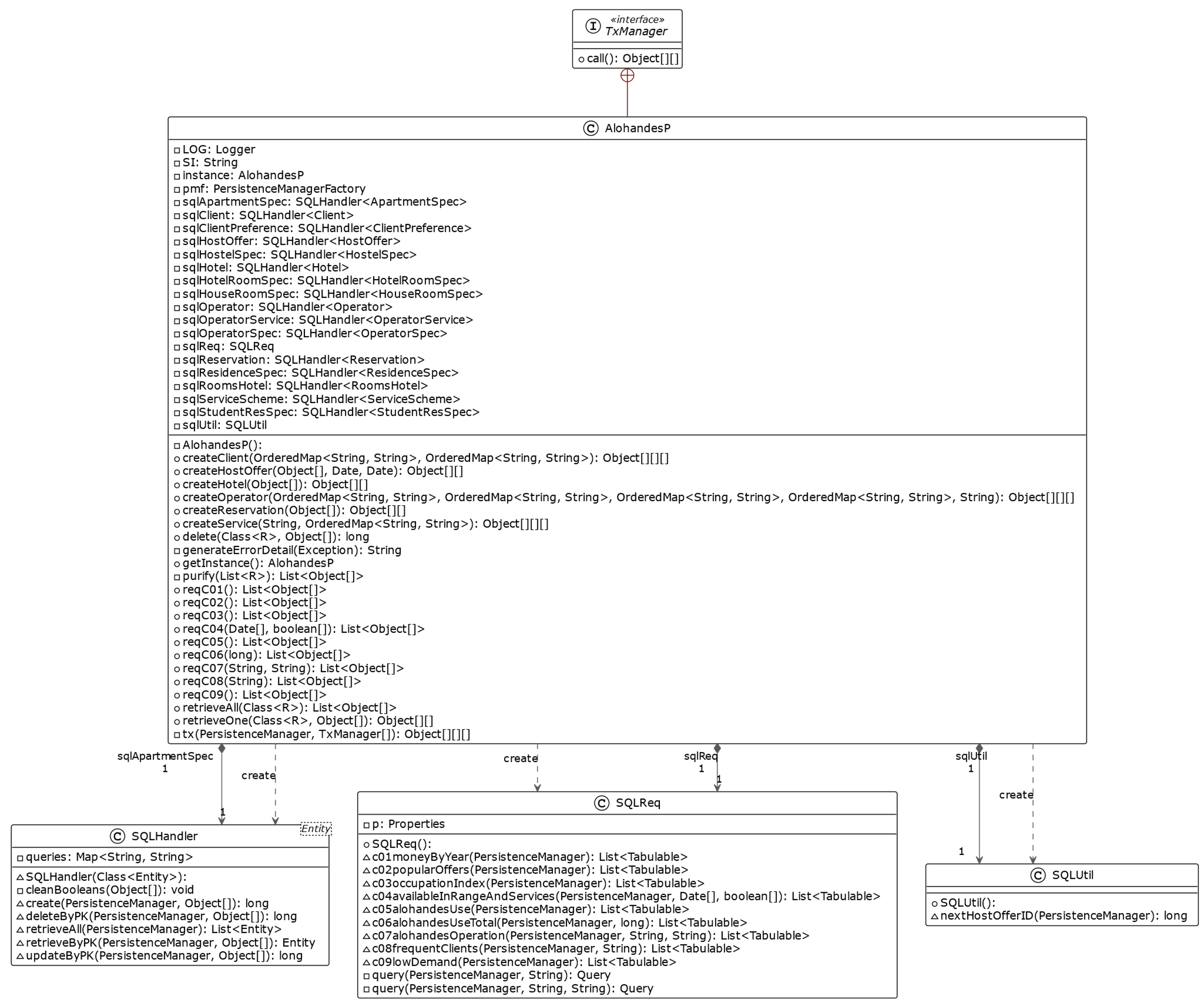
Para el desarrollo de una aplicación eficiente y escalable, es importante definir una arquitectura de solución clara y organizada. En este sentido, se ha dividido el proyecto en seis submódulos que se encargan de distintas tareas en el proceso de desarrollo:

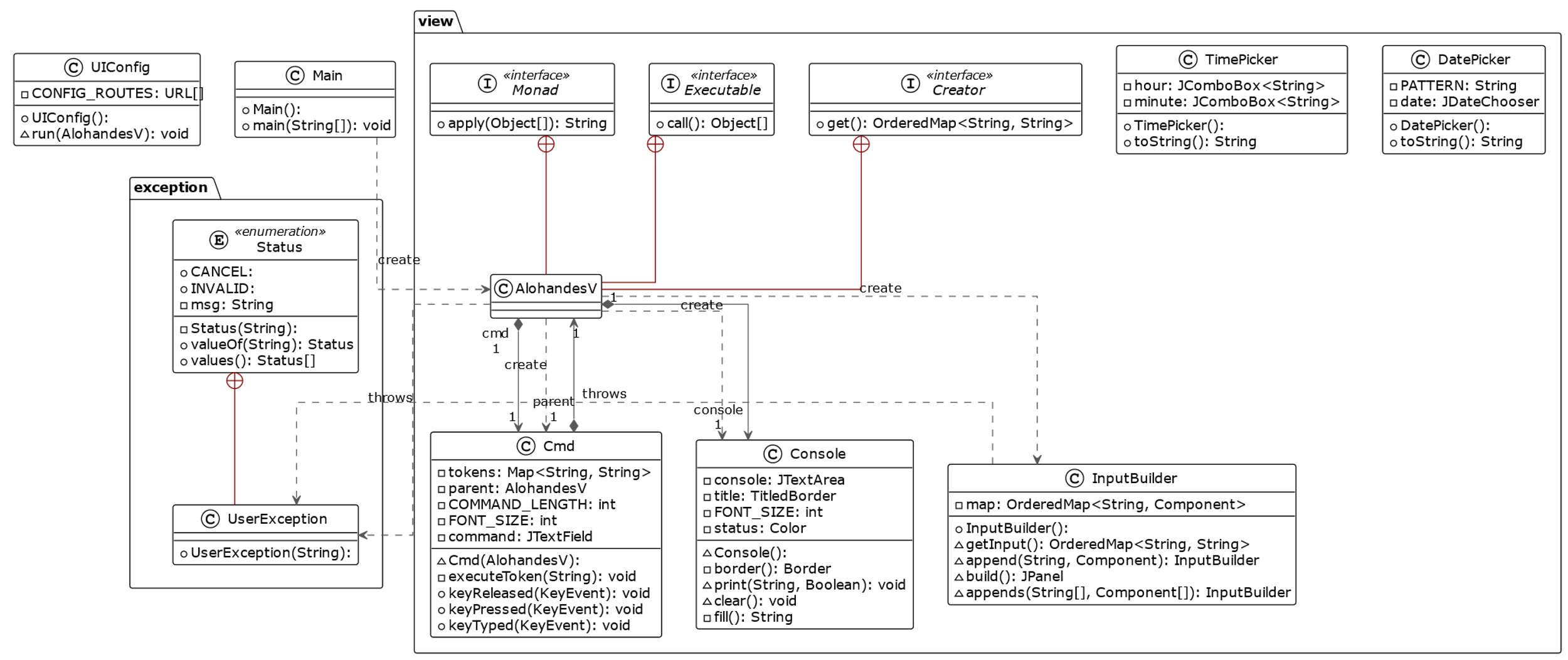
El primer submódulo, "annotation-magic", contiene un annotation-processor que facilita la creación de POJOs, VOs y Queries del CRUD mediante el uso de anotaciones.

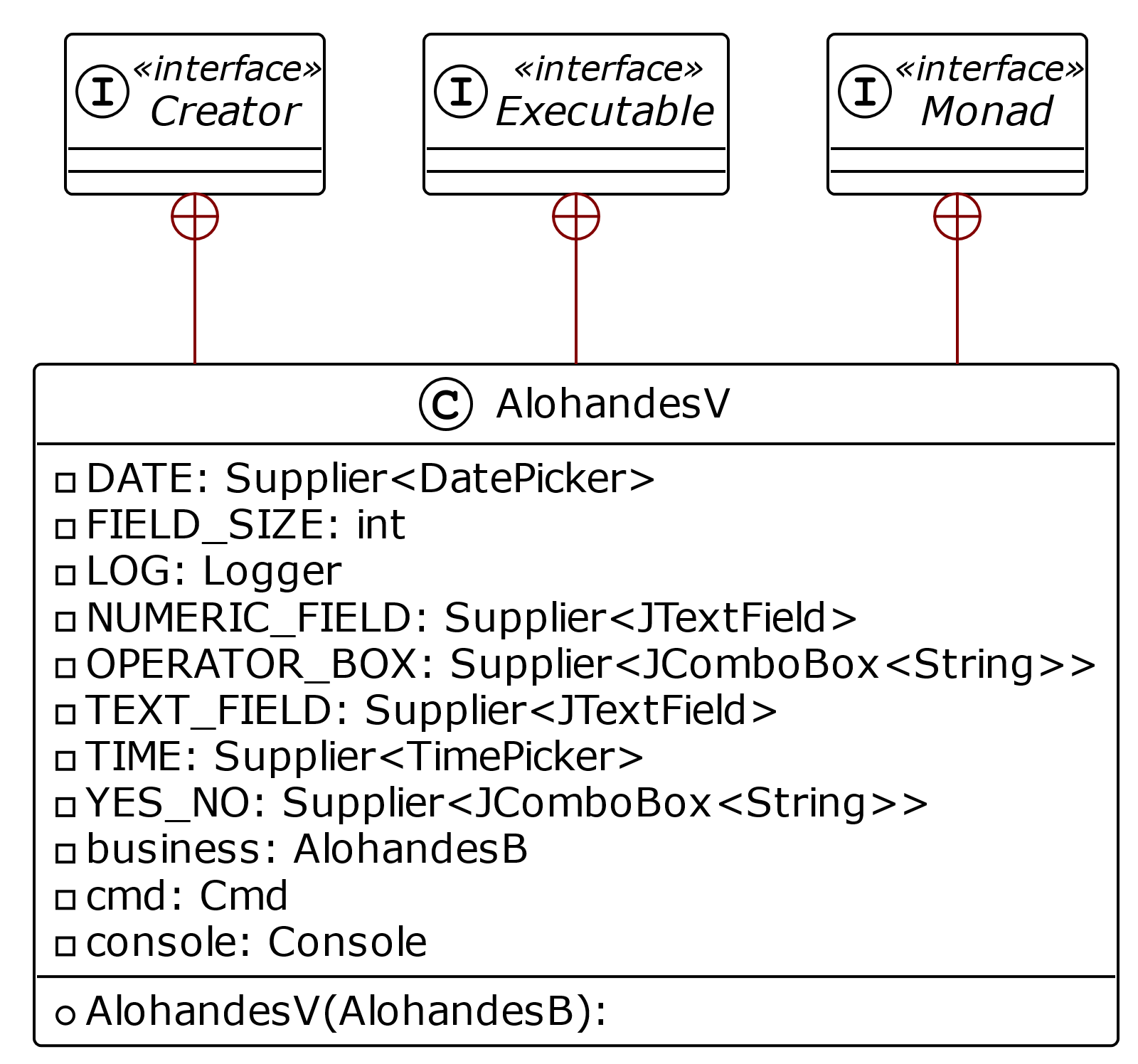


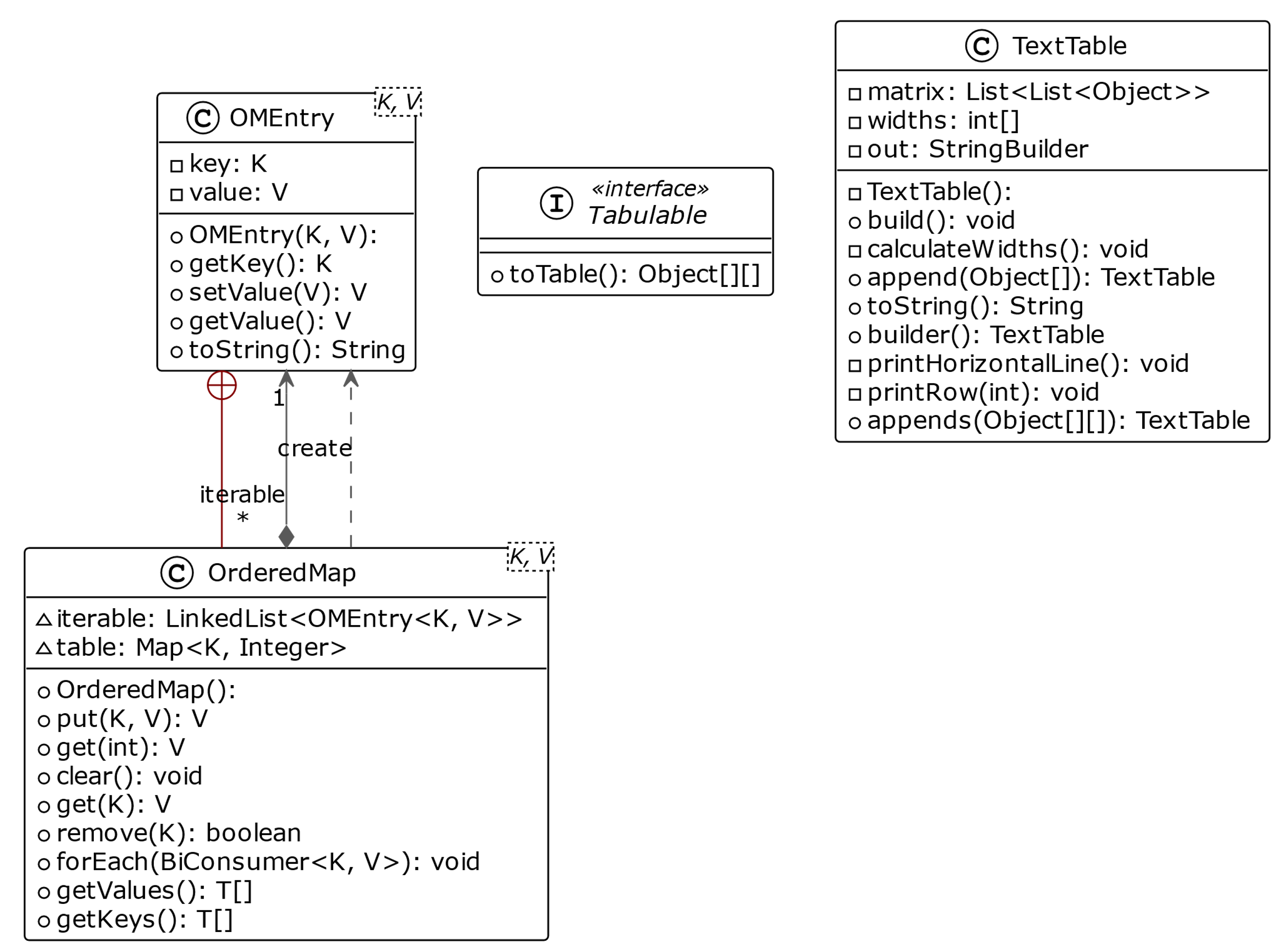
El segundo submódulo, "api", corresponde a la capa de negocio de la aplicación, donde se realizan las operaciones transaccionales y de consulta. 

El tercer submódulo, "data", se encarga de contener los POJOs y VOs, así como los objetos de acceso a datos necesarios para la persistencia en la base de datos. 

El cuarto submódulo, "persistence", se encarga de la capa que utiliza las queries y las operaciones de base de datos, garantizando la coherencia y consistencia de los datos almacenados.  
 

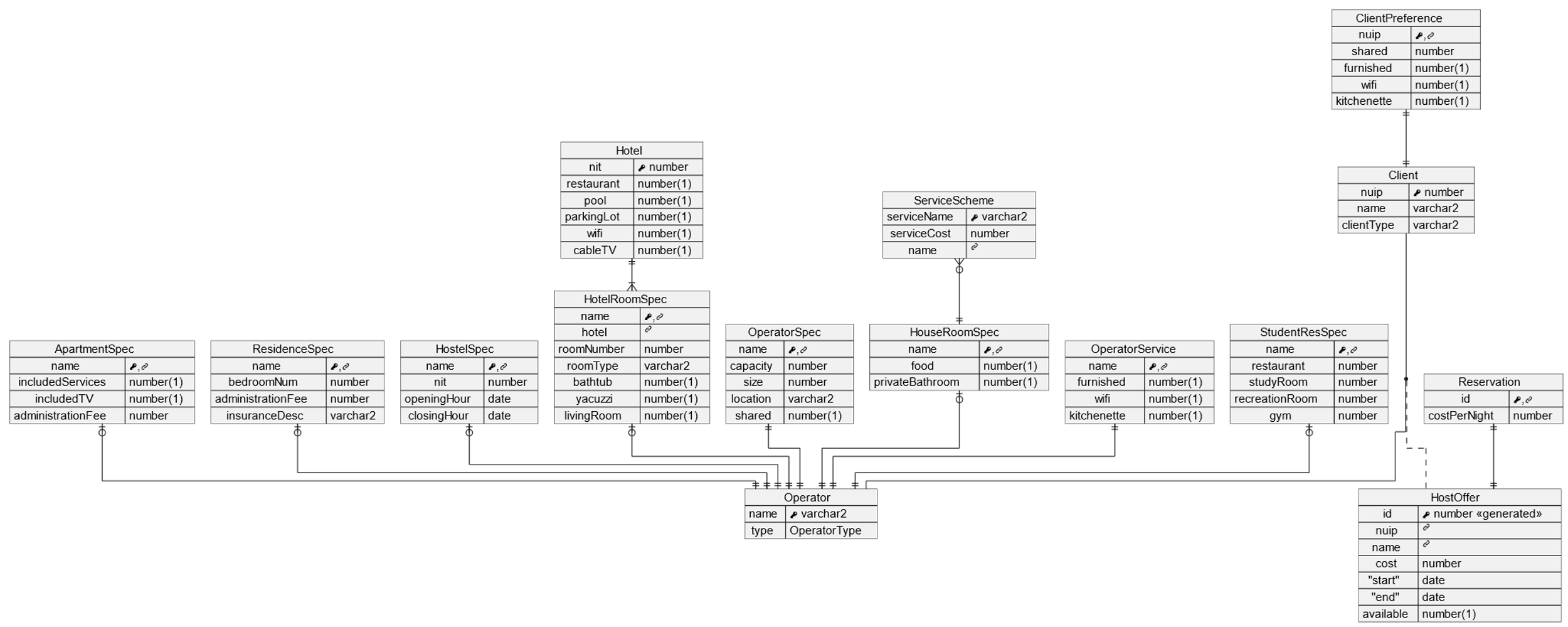
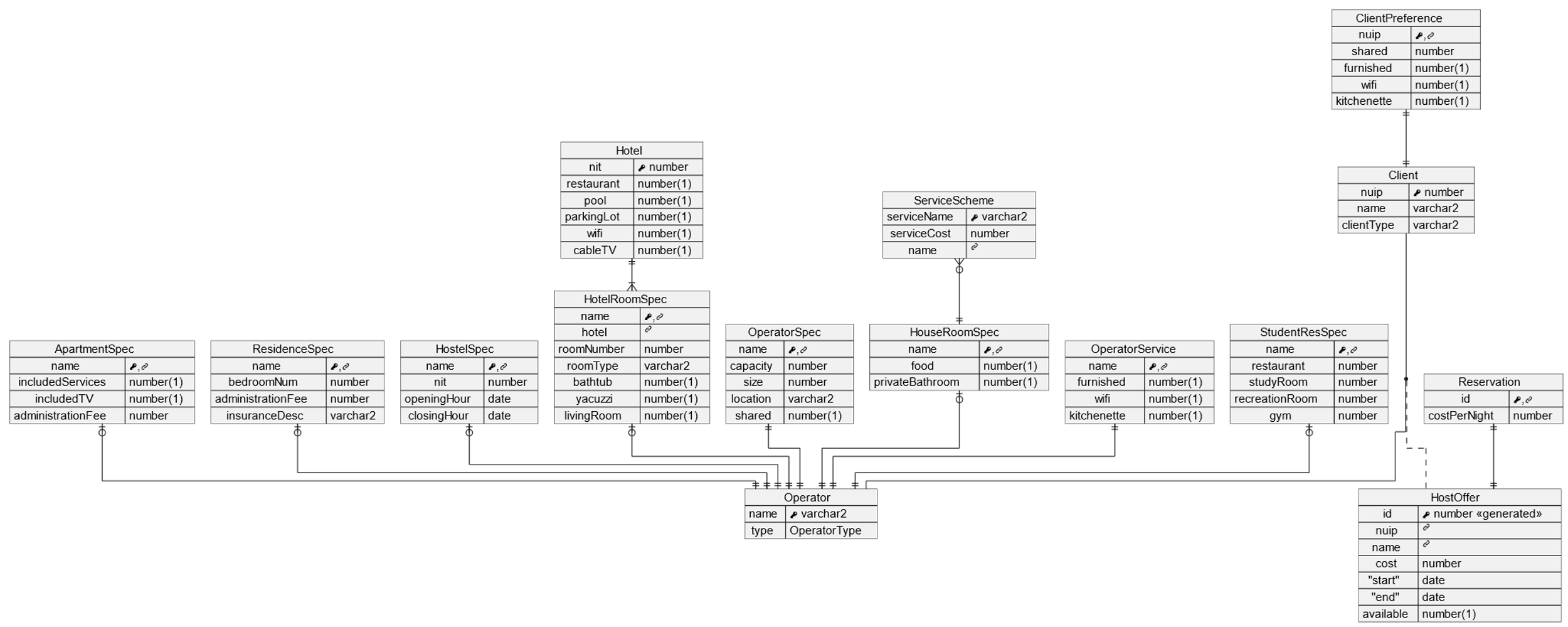
El quinto submódulo, "ui", se encarga de la interfaz de usuario, donde se realizan las operaciones de entrada y salida de datos, y se muestran los resultados de las consultas y operaciones realizadas.  
Vista general del submodulo: 

Vista de AlohandesV (Campos y constructor / El resto de metodos):  


Por último, el submódulo "utils" contiene una serie de clases que otorgan utilidad y facilitan las operaciones para las otras capas. 

Cada submódulo se ha diseñado y desarrollado de manera independiente, pero trabajando en conjunto para cumplir con los requerimientos funcionales y de consulta del proyecto. A continuación, se presentarán seis diagramas UML que describen cada uno de estos submódulos en detalle.

## Modelo relacional



# Resultados

A continuación, se muestra el funcionamiento de la aplicación y cada requerimiento (Con información cargada previamente para los de consulta)

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer screen

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence