

TRABAJO INTEGRADOR

Contexto

La empresa GoGoTravel es una agencia de viajes internacional y a comienzos del año por razones de gestión, el C-Level de la empresa tomó la decisión de actualizar los sistemas de generación de paquetes de viaje y reservación.

El equipo de análisis y diseño ha trabajado el último trimestre en conjunto con un equipo de delegados de la empresa para identificar las necesidades y requerimientos que deben satisfacerse con la producción de una nueva aplicación de la que el equipo de desarrollo se hará a cargo.

En primera medida, se acordó que el desarrollo no incluirá la construcción de un frontend, sino que deberá exponer una API REST.

Requerimientos técnicos

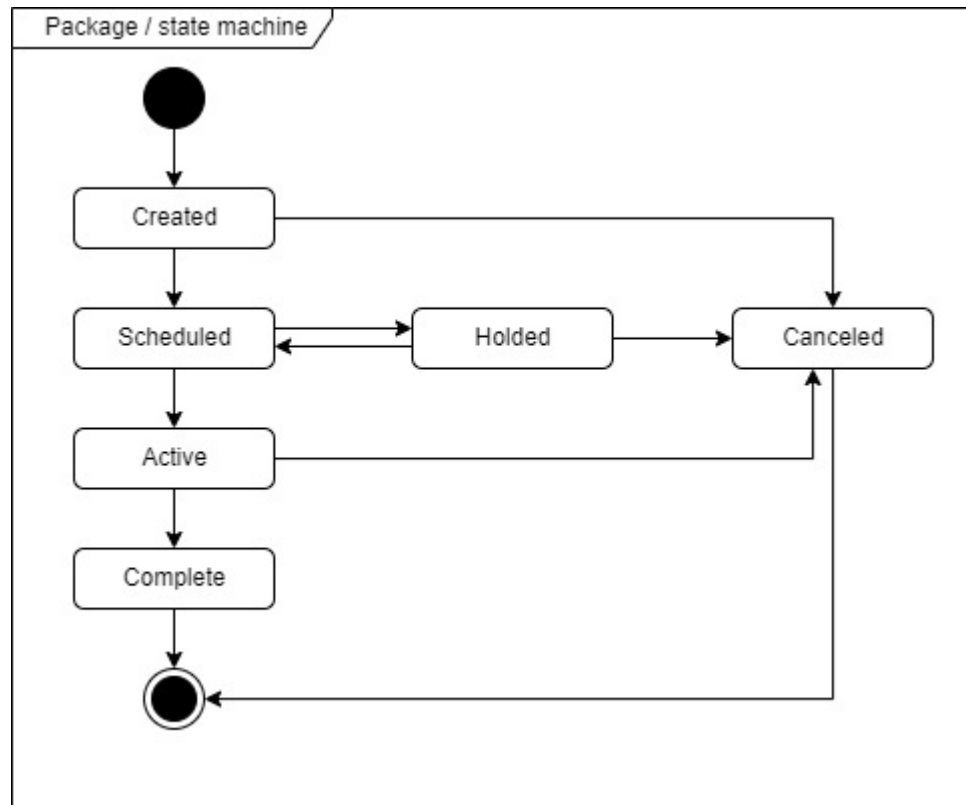
El referente técnico del equipo dispuso que los desarrolladores deberán utilizar Python en su versión 3.x como lenguaje de programación sobre el que se construirá el proyecto. A su vez los desarrolladores deberán:

- Hacer uso del gestor de paquetes *pip* para instalar las dependencias que correspondan.
- Generar un entorno virtual a través del módulo *venv* y declarar las dependencias del proyecto en un archivo *requirements.txt*
- Utilizar el ORM *sqlalchemy* para acceder a la base de datos
- Utilizar la librería FastAPI para crear una API REST que exponga los métodos necesarios para cubrir las necesidades dispuestas en el apartado *Requerimientos Funcionales*.
- Utilizar el módulo *unittest* de python para realizar pruebas unitarias de código.
- Crear a partir de Postman una colección de requests que permitan ejecutar los métodos de la API.
- El código deberá ser subido a un repositorio en Github.

Requerimientos funcionales

- Permitir la creación, modificación, eliminación y recuperación de los datos de un paquete de viajes.
- Permitir que un paquete de viaje pueda ser simple o compuesto.
- Un paquete simple (o *servicio*) debe ser de uno de los tres tipos siguientes: viaje, excursión o alojamiento.
- Un servicio de tipo "viaje" debe tener asociado un vehículo: avion, crucero, autobus o tren.
- Un servicio de tipo "excursión" debe tener asociado una excursión.
- Un servicio de tipo "alojamiento" debe tener asociado un hotel.
- Los vehículos, excursiones y hoteles, son provistos por alguna compañía prestadora, asociada a GoGoTravel.
- Un paquete compuesto puede incluir más de un *servicio*.
- Un paquete debe tener uno de los siguientes estados en todo momento:
 - **creado**: estado inicial de un paquete.

- **programado**: cuando se configura la fecha de inicio/fin del paquete.
- **activo**: cuando la fecha de inicio es la fecha actual.
- **finalizado**: cuando la fecha de fin es la fecha actual (y el estado anterior es activo).
- **cancelado**: cuando ocurre algún inconveniente que no permite volver a reprogramar el paquete.
- **espera**: cuando ocurre algún inconveniente corregible y sí se permite volver a reprogramar el paquete.



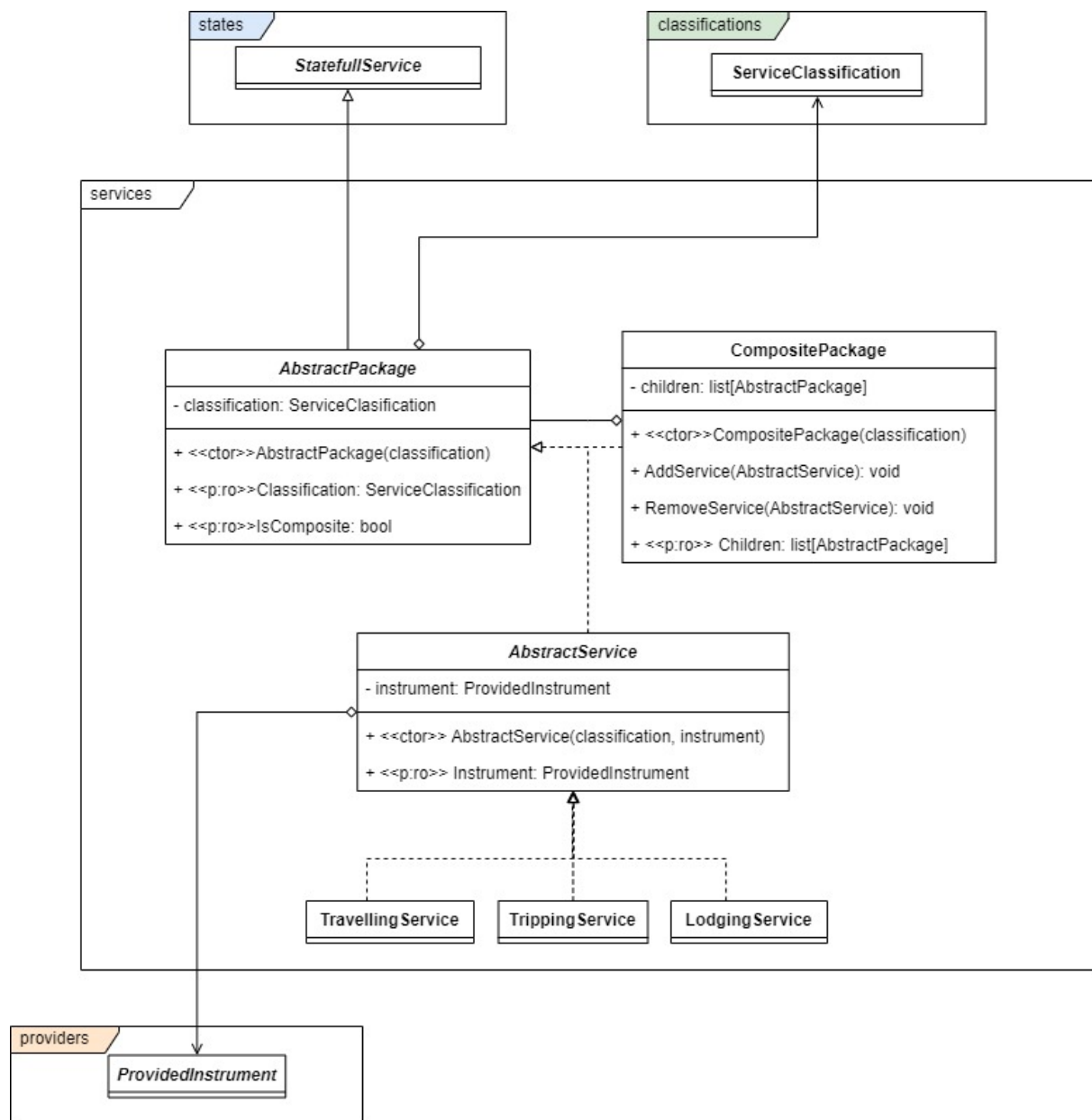
- Un paquete tiene una clasificación que se define al momento de su creación y puede ser:
 - **clase económica**: el paquete tiene un recargo del 5% en todos los servicios incluidos
 - **clase ejecutiva**: el paquete tiene un recargo del 15% en todos los servicios incluidos
 - **primera clase**: el paquete tiene un recargo del 45% en todos los servicios incluidos
- Todos los servicios de un paquete compuesto que sean de una clasificación particular, deben ser de la misma clasificación. Es decir, para un paquete de primera clase, cualquier servicio de viaje, excursión o alojamiento, deberán ser también de primera clase.

Anexos

Paquete "services"

El paquete services cuenta con la definición de las clases núcleo de la solución, correspondientes a los paquetes y servicios de viaje.

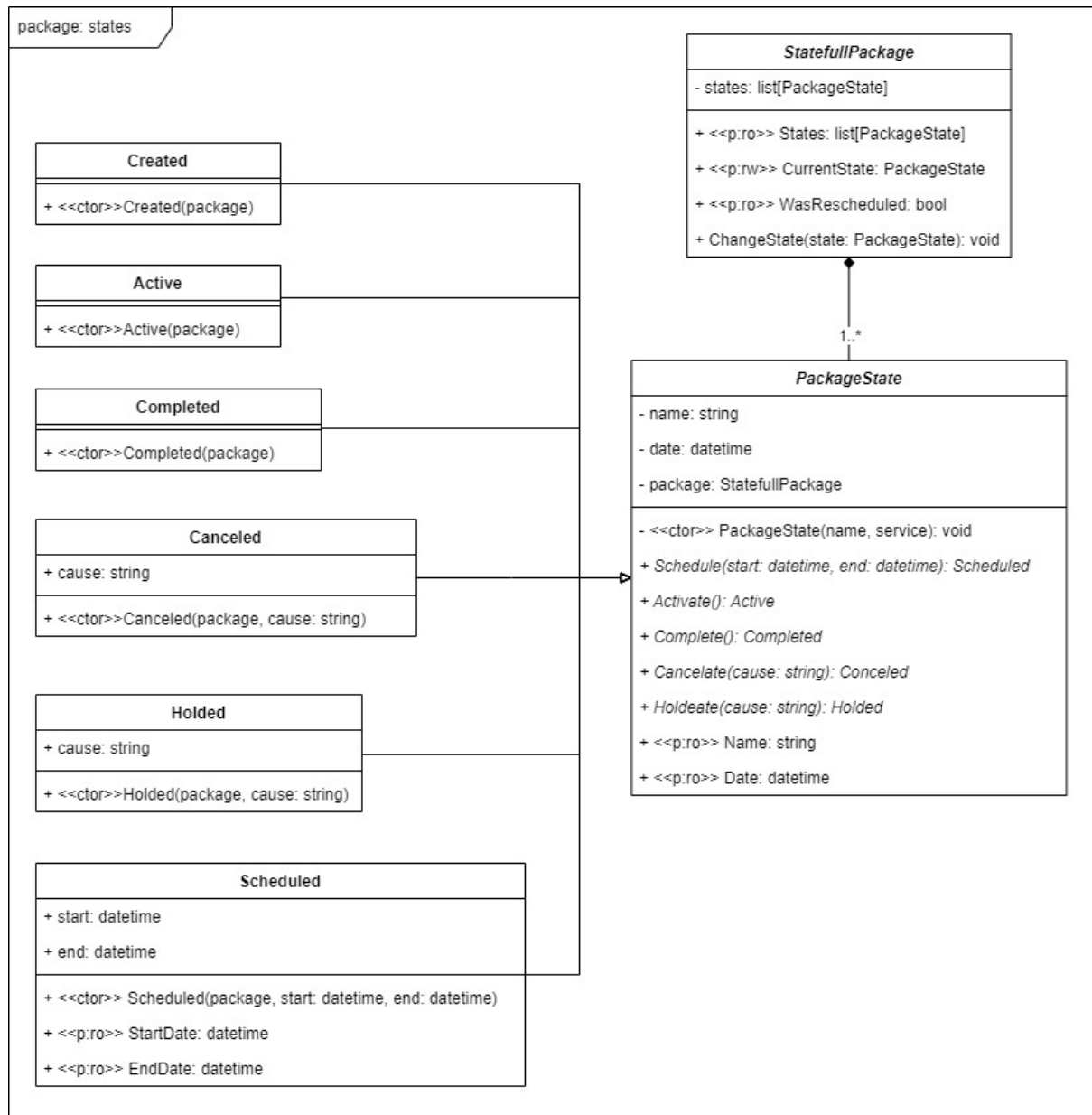
Se modelan las relaciones y dependencias con demás clases y paquetes como "states", "classifications" y "providers".



Paquete "states"

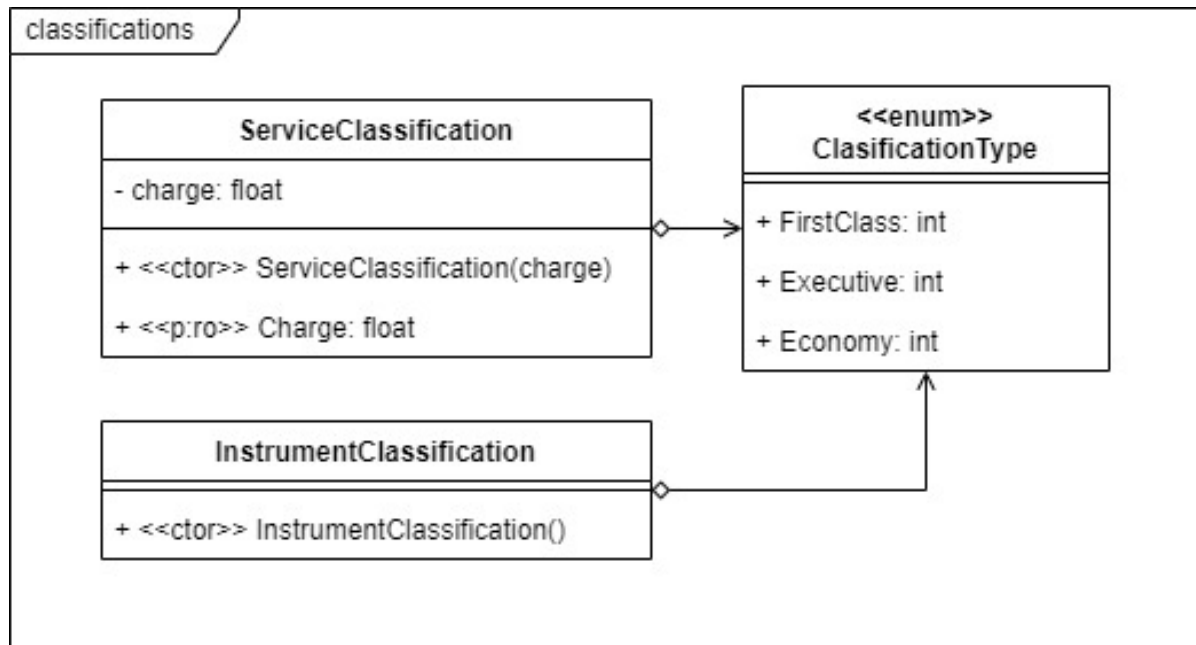
Dentro de este paquete se resolverán los estados correspondientes a un paquete de viaje. La clase *StatefullPackage* es una interfaz que será implementada por la clase *services.AbstractPackage* (ver paquete "services").

Los estados implementarán los métodos heredados de *PackageState* y respetando el diagrama de estados presentado en los requerimientos funcionales, actuarán en el cambio de estado o no harán nada.



Paquete "classifications"

El paquete classifications cuenta con la definición de las clasificaciones de los paquetes y servicios. Como se hace mención en los requerimientos funcionales, cada una de las clasificaciones tiene un recargo que se inicializa en el constructor para cada clase concreta, respectivamente.



Paquete "providers"

Este paquete define a los proveedores de servicio, empresas que cuentan con los instrumentos para viajar, realizar excursiones y/o alojarse en un hotel.

Para estos instrumentos se necesitan algunos datos, como por ejemplo:

- Un viaje tiene los datos del vehículo: tipo, matrícula, etc.
- Una excursión tiene datos del guía: nombre, legajo y teléfono.
- Un hotel necesita los datos de la dirección.

