

# Parcial Práctico 1

## Instrucciones

1. Clone el repositorio
2. Abra el proyecto en Visual Studio Code (o en el editor de su preferencia)
3. Lea el enunciado completamente antes de iniciar

## Punto 1 (30%). Persistencia

Esta aplicación tiene el propósito de crear un sistema de información para una cadena de hoteles.

(10%) Cree la entidad HotelEntity. Un Hotel tiene un nombre, una dirección, un número de estrellas y un id de tipo Long que representa su llave primaria.

(10%) Cree la entidad HabitacionEntity. Una habitación tiene un número de identificación, un número de personas que pueden estar en la habitación, un número de camas, un número de baños y un id de tipo Long que representa su llave primaria.

Tenga en cuenta que un hotel tiene un número de habitaciones y una habitación solo está en un hotel.

(10%) Cree los repositorios para HotelEntity y HabitacionEntity.

## Punto 2 (30%). Lógica

(10%) Cree la clase correspondiente para la lógica de Hotel. Implemente únicamente el método createHoteles. Valide que el nombre del hotel no esté repetido y que el número de estrellas este entre 1 y 5.

(10%) Cree la clase correspondiente para la lógica de Habitación. Implemente únicamente el método createHabitacion. Valide que una habitación  $\text{numeroIdentificación} \leq \text{numeroCamas}$ .

(10%) Cree la clase correspondiente para la lógica de la asociación entre Hotel y habitación. Implemente únicamente el método addHabitacion. Este método recibe como parámetro el id del hotel, el id de la habitación y le agrega la habitación al hotel. Valide que tanto el hotel como la habitación existan.

## Punto 3 (40%). Prueba de lógica

(10%) Implemente las pruebas para el método `createHoteles` del servicio de hotel. Asegúrese de crear dos pruebas: una donde el hotel se crea correctamente y otra donde se lanza una excepción de negocio por la violación de la regla de negocio.

(10%) Implemente las pruebas para el método `createHabitacion` del servicio de habitación. Asegúrese de crear dos pruebas: una donde la habitación se crea correctamente y otra donde se lanza una excepción de negocio por la violación de la regla de negocio.

(20%) Implemente las pruebas para el método `addHabitacion` del servicio de la asociación. Asegúrese de crear tres pruebas: una donde la habitación se agrega correctamente al hotel; otra donde se lanza una excepción de negocio porque el hotel no existe; y otra donde se lanza una excepción porque la habitación no existe.

## **Punto 4 (10%). Bono API**

- Cree las clases `HabitacionDTO` con los atributos correspondientes.
- Cree la clase `HabitacionController`
- Implemente el método de la logica que crea una habitación.
- Cree una prueba de Postman que verifique que se crea la habitación de forma correcta

Solo se obtiene el bono si se realiza de forma completa, hacer parte del bono y no terminarlo no da puntajes intermedios.