

# ISW1-Sims Hotels

En la próxima conquista espacial, muchos están ocupados construyendo naves espaciales, pero solo una empresa tiene como objetivo abrir el primer hotel fuera de la Tierra. Y es por esto que nos contrataron para implementar una simulación para mejorar su servicio.

Ya tenemos la primera implementación del modelo (implementado por otro equipo) con \*deuda técnica\* que queremos abordar en esta primera iteración.

## Modelo

Un **Hotel** es una colección de pisos (que llamaremos **Floor**) y a su vez cada **floor** tendrá habitaciones (que llamaremos **Room**).

El **Hotel** podrá:

- Recibir (sin previa reserva) a un **guest** en un **room** de un **floor** (*Hotel>>#receive:atFloor:atRoom:.*).
- Tomar la reserva de un **guest** (*Hotel>>#reserveRoom:atFloor:.*).
- Recibir un **guest** en un **room** reservado, que a diferencia de un **room** disponible, solo puede ser accedido con el mensaje *Hotel>>#receiveWithReservation:atFloor:atRoom:.*

Notar que para esta iteración (y para simplificar el problema), no se verifica que un **room** reservado sea accedido por el mismo **guest** que hizo la reserva, es más, es importante también notar que **los guests están modelados a partir del tipo de guest usando símbolos:** pueden ser un **#conferenceGuest** o un **#vacationGuest** (ver en los tests los mensajes **#guestTypeConference** y **#guestTypeVacation**).

¿Para qué sirve saber el tipo de **guest** que ocupa cada habitación?

Cada **floor** tiene un precio asociado diferenciado por el tipo de **guest**. Esto sirve para calcular cuál es la ganancia (y pérdida) del **hotel** (*Hotel>>#totalProfits* y *Hotel>>#totalLosses*)

## Trabajo a realizar

Con todo esto en mente, nos piden:

1. Arreglar la instanciación de **Floor**. Luego modificar los mensajes *#createFloorWith:and:* en los tests.
2. Arreglar la creación de **Hotel**. Luego modificar los mensajes *#createHotel* en los tests.
3. Modelar el hecho de que una habitación puede estar ocupada, disponible o reservada de una manera más explícita y declarativa. El equipo anterior parece haber utilizado el **guest** con **nil** para modelar una habitación disponible, y el tipo de guest *#reserved* para la habitación reservada ...
4. Una vez arreglado el ítem 3, que no quede código repetido entre los métodos *#receive;*, *#receiveWithReservation* y *#reserve*.

5. Hemos notado que en varios métodos de **Hotel** se utiliza **floor rooms** rompiendo el encapsulamiento. ¡Arreglemos esto! (Hint: faltan mensajes en **Floor**).
6. Relacionado al ítem 5, hacer funcionar los tests 13, 14 y 15 de **Room**. Actualmente están implementados usando el mensaje **#shouldFail**. Borrar dicho mensaje de cada test y hacer pasar lo que está dentro del bloque. Agregar el mensaje que falta en **Room** y arreglar *Hotel>>#totalLosses* (Hint: ver la implementaciones de *Floor>>#totalProfits* y de *Room>>#profitUsing:ifAbsentGuestType*).
7. ¡Hemos encontrado mucho **código repetido**! Nos piden mejorar los siguientes mensajes si aún no lo has hecho:
- *Hotel>>#receive:atFloor:atRoom*., *Hotel>>#receiveWithReservation:atFloor:atRoom*: y *Hotel>>#reserveRoom:atFloor*..
  - *Hotel>>#totalLosses* y *Hotel>>#totalProfits* (¿existe algún mensaje de colecciones que también nos ayude a mejorar la declaratividad de estas implementaciones?)
  - *Hotel>>#isEmpty*, *Hotel>>#totalRooms*, *Hotel>>#totalRoomsOccupied* y *Hotel>>#totalRoomsReserved* (primero mejoremos la declaratividad de estos métodos!).
8. Bonus Track: Solo si terminaste todo lo anterior, tenés tiempo y energía, mejorar los tests evitando que rompan en encapsulamiento y sacando el código repetido que consideren necesario. Esto solo suma puntos, no baja puntos si no lo hacen.

## Entrega:

1. Entregar por mail el fileout de la categoría de clase **ISW1-2024-1C-1erParcial** que debe incluir toda la solución (modelo y tests). El archivo de fileout se debe llamar: **ISW1-2024-1C-1erParcial.st**
2. Entregar también por mail el archivo que se llama **CuisUniversity-nnnn.user.changes**
3. Probar que el archivo generado en 1) se cargue correctamente en una imagen “limpia” (o sea, sin la solución que crearon. Usen otra instalación de CuisUniversity/imagen si es necesario) y que todo funcione correctamente. Esto es fundamental para que no haya problemas de que falten clases/métodos/objetos en la entrega.
4. Realizar la entrega enviando mail a la lista de Docentes: [docentes@isw2.com.ar](mailto:docentes@isw2.com.ar) con el **Subject: LU nnn-aa - Solución 1er Parcial 1c2024**
5. **RECOMENDACIÓN IMPORTANTE: Salvar la imagen de manera frecuente o con el autosave**
6. Se asume que a esta altura de la cursada saben trabajar con la imagen, recuperarla, revertir cambios y demás incidencias que pudieran ocurrir durante el examen.

**Revisen bien los puntos de arriba. Cualquier error en los nombres o formato podrían ser penalizados en la nota.**

**IMPORTANTE: No retirarse sin tener el ok de los docentes** de haber recibido el mail con la resolución.

**LAS ENTREGAS RECIBIDAS DESPUES DE LAS 22:00 HRS NO SERAN TENIDAS EN CUENTA**