

## Objetivo general del proyecto

El objetivo general de este proyecto es practicar varias etapas del desarrollo de una aplicación de software, desde el análisis hasta la construcción una aplicación funcional. A través del proyecto, los estudiantes pondrán en práctica todas las habilidades desarrolladas en el curso.

## Objetivos específicos del proyecto

Durante el desarrollo de este proyecto se buscará el desarrollo de las siguientes habilidades:

1. Identificar y abstraer entidades en un dominio particular, así como sus características y relaciones.
2. Construir diagramas de clase UML para expresar modelos de dominio y modelos de diseño.
3. Diseñar una aplicación basada en el paradigma orientado a objetos y expresar el diseño, justificando las decisiones importantes que hayan tomado.
4. Implementar un diseño utilizando el lenguaje de programación Java.

## Instrucciones generales

A continuación, encontrará la definición inicial del proyecto, que será la base del trabajo de los 3 proyectos que se construirán durante todo el semestre.

El trabajo del proyecto #2 debe partir de la base del proyecto #1, y el trabajo del proyecto #2 será la base del proyecto #3. Si es necesario, podrán hacerse cambios en cada una de las etapas para mejorar o corregir decisiones que se hayan tomado en las etapas anteriores.

En el proyecto 1 no es necesario que haya una interfaz gráfica: la implementación que realicen puede tener una interacción con el usuario basada en consola.

El proyecto debe desarrollarse en **grupos que podrán cambiar para el proyecto #2, pero no para el proyecto #3.**

## Contexto del proyecto

Dentro del proyecto del curso vamos a construir un sistema que los hoteles llaman “Property Managament System” o PMS, el cual se encargará de administrar muchos de los elementos más importantes del hotel.

El PMS se encargará de tres grandes funcionalidades. En primer lugar, el sistema tendrá un inventario de las habitaciones del hotel. En segundo lugar, el sistema tendrá el catálogo de servicios ofrecidos en el hotel y un registro de consumo. En tercer lugar, el sistema debe ser capaz de manejar todos los aspectos relacionados con las reservas, los huéspedes y los pagos.



## Inventario

El sistema debe ser capaz de manejar un inventario de habitaciones <sup>1</sup> para el hotel que serán utilizados para las reservas. Cada habitación tendrá un identificador (número o nombre), una ubicación e información sobre su capacidad. El hotel tiene habitaciones tipo estándar, suite, y suite doble. Independientemente del tipo, algunas habitaciones tienen balcón, algunas tienen vista y otras tienen cocina integrada. Para las camas que aparecen en la descripción de cada habitación es importante saber el tamaño y la cantidad de personas que caben en ellas. Por su tamaño, algunas camas sólo pueden ser utilizadas por niños.

El sistema debe permitir que el administrador del sistema cree las habitaciones una por una, o cargue un archivo con la información de nuevas habitaciones (o información actualizada de habitaciones existentes).

Los empleados de la recepción del hotel podrán consultar el inventario y las características de las habitaciones, para saber qué características tienen, quién las está ocupando en un determinado momento, y quién las tiene reservadas en una determinada fecha.

El administrador del sistema también puede cargar las tarifas por tipo de cuarto, las cuales serán válidas en determinadas fechas. En el momento de configurarlas, el administrador especificará el tipo de cuarto (ej. suites), un rango de fechas (ej. entre 10 de enero y 18 de marzo) y unos días de la semana (ej. viernes, sábado y domingo) para los cuales aplicará una tarifa. Si el administrador se equivoca y aplica más de una tarifa para el mismo tipo de habitación para la misma fecha, el sistema debe tomar la tarifa más económica. El sistema debería avisarle al administrador si hay alguna fecha dentro de los próximos 365 días en la que no exista una tarifa asignada para un cierto tipo de habitación.

El administrador también puede cambiar las tarifas de los servicios que ofrezca el hotel, cargar los menús del restaurante y configurar todo lo que sea necesario de cada plato.

## Servicios

El sistema debe ser capaz de registrar los servicios que hayan sido consumidos por un huésped, registrar los pagos que se hayan hecho, y generar las facturas correspondientes. La mayoría de consumos se pueden pagar inmediatamente o se pueden registrar a una habitación para que el huésped sólo tenga que pagar en el momento de hacer el registro de salida (check-out) <sup>2</sup>. Los consumos serán registrados en el sistema por un empleado del hotel (ej. mesero, encargado del spa).

El principal servicio que presta el hotel es el de alojamiento: cualquier huésped del hotel debe estar alojado en una habitación, y cada habitación generará un cobro (se paga por habitación, no por persona). Los huéspedes del hotel pueden utilizar otros servicios que ofrezca el hotel, como por ejemplo el uso del spa (que se cobra por persona) o el servicio de guía turístico (que se cobra por grupo).

Los servicios de restaurante que presta el hotel pueden ser tomados en el comedor o en la habitación, pero no todas las opciones del menú pueden llevarse a la habitación. El sistema debe tener información completa sobre el menú, incluyendo los nombres de los platos y bebidas, los precios, y los rangos de horas en los que están disponibles.

## Reservas, registro y facturación

El sistema debe permitir que un empleado de la recepción del hotel realice una reserva a nombre de un huésped, y le informe la tarifa total para las noches seleccionadas. Las reservas también pueden cancelarse, pero no en las

<sup>1</sup> En esta sección, le recomendamos revisar un servicio de reserva como booking.com para buscar inspiración.

<sup>2</sup> Su sistema no debe hacer nada con el pago además de registrar que se haya hecho. Por ahora no es necesario que verifique datos de una tarjeta, se conecte a una pasarela de pagos ni nada similar.

# Archivo Tarifas

1/1	Tarifa
2/1	+ 5000
3/1	
⋮	
⋮	

⇒ 4 archivos

habita suit

últimas 48 horas. Para hacer una reserva se necesita el nombre del huésped, documento, correo electrónico, número de celular y la cantidad total de personas que se esperan. Al hacer una reserva, las habitaciones necesarias quedarán bloqueadas en las fechas de la reserva.

En el momento de hacer el registro, el empleado de la recepción le pedirá al huésped los datos básicos de él y de sus acompañantes para registrarlos en el sistema. Un huésped y sus acompañantes se consideran un grupo y pueden tener una o varias habitaciones asignadas. Un grupo siempre debe registrarse y hacer el registro de salida al tiempo, después de haber pagado por todos sus consumos y por la estadía. El límite de personas por habitación depende de la capacidad de las camas, pero los menores de dos años no necesariamente necesitan cama –es decisión de los padres- aunque siempre deben quedar registrados dentro de los huéspedes.

La recepción podrá generar las facturas para todos los consumos realizados durante la estadía de un grupo en el hotel.

### Aspectos técnicos y restricciones

Toda la información debe ser persistente (el inventario, los servicios, menús, reservas, consumos, etc.). La información debe almacenarse en archivos (pueden ser planos o binarios), dentro de una carpeta y se puede suponer que sólo la aplicación va a escribir y leer de esa carpeta (ningún usuario malicioso va a modificar los archivos que ahí se encuentren sin utilizar la aplicación). La carpeta no puede ser la misma carpeta donde se encuentre el código fuente de la aplicación.

La persistencia no necesariamente debe hacerse en un solo archivo: diseñe con cuidado cuántos archivos habrá y cómo van a estar estructurados.

Los usuarios del sistema serán el administrador, los recepcionistas y los otros empleados del hotel que registren servicios y consumos. Todos los usuarios del sistema deben tener un login y un password y usarán la misma aplicación: dependiendo del tipo de usuario, las opciones que se muestren deben ser diferentes.

Para fines de auditoría, debe ser posible generar un archivo de log con el historial de un grupo de huéspedes.

La aplicación debe estar hecha en Java y la interfaz debe estar basada en consola.

No es necesario que se soporten funcionalidades que no hayan sido mencionadas, pero se pueden incluir funcionalidades adicionales si facilitan el trabajo.

## Entrega 1: Análisis del proyecto

La primera etapa del proyecto consiste en realizar el análisis del sistema que debe construir.

### Actividades

1. Construya un modelo de dominio (diagrama de clases) a partir de la información del caso:
  - a. Identifique las entidades que aparecen dentro del caso.
  - b. Identifique las características (atributos) de esas entidades.
  - c. Establezca las relaciones entre las entidades, incluyendo asociaciones y relaciones de herencia.
2. Construya un documento de análisis para su proyecto. El documento debe incluir:
  - a. El modelo de dominio que construyó en el paso anterior.
  - b. Una descripción de los requerimientos funcionales que su aplicación soportará. Le recomendamos utilizar historias de usuario.
  - c. Una descripción con las restricciones del proyecto.

## Entrega

1. El proyecto debe entregarse en una carpeta dentro del repositorio GIT del grupo con el nombre **"Proyecto 1"**. Dentro de esta carpeta debe existir una carpeta con el nombre **"Entrega 1"** donde deben quedar todos los elementos correspondientes a esta entrega, incluyendo tanto los archivos fuente de los diagramas como imágenes que se puedan leer con facilidad.
2. Entregue un enlace al repositorio a través de Bloque Neón en la actividad designada como **"Proyecto 1 - Entrega 1"**.

## Entrega 2: Diseño

Teniendo en cuenta el análisis realizado en la primera entrega del proyecto, para la segunda entrega debe realizar el diseño detallado del sistema.

### Actividades

1. Realice el diseño y construya un documento de diseño donde presente el diseño, **haciendo especial énfasis en justificar las decisiones clave que haya tomado**. El documento debe incluir, como mínimo, los siguientes elementos:
  - a. Un diagrama de clases de diseño que incluya todas las clases, incluyendo sus las relaciones, atributos y métodos.
  - b. Un diagrama de clases de alto nivel, que incluya todas las clases y sus relaciones, pero no todos los métodos ni atributos. Este diagrama facilitará entender las relaciones entre clases.
  - c. Diagramas de secuencia para las funcionalidades que usted considere críticas:

Estos elementos NO son los únicos que debe incluir su documento: piense en qué otros diagramas, descripciones y justificaciones pueden ser necesarias para poder entender totalmente el diseño. Si su documento incluye únicamente los tres elementos mencionados antes, con seguridad será considerado insuficiente.

## Entrega

1. Dentro de la carpeta del proyecto debe crear una carpeta con el nombre **"Entrega 2"** donde deben quedar todos los elementos correspondientes a esta entrega, incluyendo tanto los archivos fuente de los diagramas como el documento de diseño.
2. Entregue un enlace al repositorio a través de Bloque Neón en la actividad designada como **"Proyecto 1 - Entrega 2"**.

## Entrega 3: Diseño e Implementación

Teniendo en cuenta el análisis y el diseño, para la tercera entrega debe realizar construir la implementación del sistema.

### Actividades

1. Si descubrió problemas con su diseño, actualícelo.



tarifas tipo de habitacion

11  
 ✓ se cargaron las tarifas de este...  
 ✓ falta una tarifa por el día tal,

Cuando se crean los objetos toca mirar si la tarifa que se va añadir en una fecha ya tiene anteriormente una tarifa. Si si quedara la más económica.

## Catalogo

habitación  
 113 - Suite, 114 - 4, 3, 12  
 Balco, Vrs, etc.

{ id: [atributo]  
 { } , ... }

{ spa: precio  
 { guca: precio }

## Menu

{ nombre: { precio, se pa llara: habitacion  
 { } , ... }

## Tarifas

lista de diccionarios.  
 cada diccionario representa un tipo de habitación. El diccionario tiene como llave el día del año ej (1/1, 12/3, 30/12) y como valor la tarifa

Cuando se vaya ordenar algo. Primero se revisa en el inventario la disponibilidad y después el catálogo para el precio y demás cosas.

## inventario

Quien cambia los valores de disponibilidad debe ser el empleado según las reservas.

✓ habitaciones

{ id: { 1/1 : { disponible: true, hasta de reserva: 1/2  
 { } , ... }

✓ menu

{ nombre: { hora: 7-5  
 { } , { } , ... }

el catalogo solo se van a getters.

(puede tener setters)

# Empleado

<<Structuror>>

Generar Factura

Realizar reserva

- ✓ INFO hosped.
- ✓ fecha de.
- ✓ fechas.
- ✓ # de habitaciones; size de hab.  
habitaciones: [10, ]  
↓  
Cocina  
Baño  
Vista

Motor.

↔ ✓ Set disponibilidad en inventario

✓ Precio total.

El empleado mira cuantos acompañantes y con eso asigna el cuarto según su capacidad ocupando la cantidad mínima de habitaciones

Servicios Consumidos

✓ nombre  
✓ precio

agregación a nuestra de tener.

Servicios: Precio

Total neto:

Iva:

Total:

Pagado: T-F

Check-in:

Acompañantes

Check-out:

Solicitar factura.

Cancelar Reserva

\* con más de 48 horas antes de la fecha llegada

✓ Set disponibilidad

✓

Cliente

→ nomb  
poc  
corre  
cel  
# de acomp.

cancelar reserva.

✓ Solicitar reserva

→ fecha //  
fecha sal.  
acom p.

huesped de  
reserva

→ info cliente  
servicios consumidos

✓ Solicitar Menu

✓ Solicitar servicio habitacion

✓ Solicitar SPA

✓ Solicitar guia

✓ Solicitar Check IN

dar info  
acompañantes

✓ Solicitar Check OUT. → paga.



2. Implemente el sistema que diseñó. Tenga en cuenta que los detalles de la implementación deben ser coherentes tanto con el modelo de clases, como con los diagramas de secuencia.
3. Asegúrese de que la documentación del diseño sea consistente con la implementación.

### Entrega

---

1. Dentro de la carpeta del proyecto debe crear una carpeta con el nombre **“Entrega 3”** donde deben quedar todos los elementos correspondientes a esta entrega, incluyendo tanto el documento de diseño actualizado como el proyecto Eclipse e instrucciones para su ejecución. Incluya también archivos de prueba para poder correr las aplicaciones y tener datos con los que se pueda probar con facilidad.
2. Entregue un enlace al repositorio a través de Bloque Neón en la actividad designada como **“Proyecto 1 - Entrega 3”**.