

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

orgInn

**DIEGO ANDRÉS CARVALLO REYES
RAFAEL IGNACIO DÍAS OYANEDEL
VICENTE TOMÁS DE JESÚS SALGADO JADUE**

Modelamiento de sistemas

Abril, 2019

Índice

Lista de Figuras	II
1 Dominio Del Problema.	1
2 Análisis.	2
3 Diseño.	3
3.1 Hotel	3
3.2 Pisos	4
3.3 Habitaciones	6
3.4 Persona	8
3.5 interfaces gráficas	9
4 Planificación.	10
5 Conclusiones.	12

Lista de Figuras

1	Clase Hotel	3
2	Clase Pisos	4
3	Clase Habitaciones	6
4	Clase Persona	8
5	ventana principal	9
6	carta gantt equipo	10
7	carta gantt Diego Carvallo	10
8	carta gantt Rafael Días	11
9	carta gantt Vicente Salgado	11

1. Dominio Del Problema.

Las grandes cadenas hoteleras ostentan de sofisticados softwares en el ámbito de la administración de habitaciones y manejo de personas en ellas. En contraparte nos encontramos con los administradores de pequeños hoteles u hostales los cuales no tienen la capacidad económica para mantener estos servicios ni tampoco están dispuestos en aprender tan compleja forma de administrar sus habitaciones. Este perfil de administrador hotelero recurrentemente utiliza softwares de dominio publico y poco especializados como Excel, o en los casos más extremos, aún utilizan libros para registrar las actividades en torno a las habitaciones.

Es por esto y motivados por el segundo grupo de administración que ofrecemos nuestro software, sencillo en su ejecución, amigable con el usuario y con todos los requerimientos que necesite sin sobresaturar de información al usuario.

2. Análisis.

para desarrollar la aplicación se necesita tener en cuenta la unidad base para la administración de habitaciones, y aunque parezca que esa unidad mínima es una habitación, la realidad es que el elemento más importante son las personas, y debido a eso es que las personas son la unidad mínima de la aplicación.

Luego de esto vendrían las habitaciones, las cuales son de diferentes tipos, con cantidades de habitaciones variando una de otras, cantidades de baños y en donde las personas se encontrarán en el hotel.

Subiendo de nivel tenemos los pisos, que son el conjunto de las habitaciones en un piso determinado, podemos sacar información sobre las mismas habitaciones, si tienen baños privados o si hay baños en ese piso, además de establecer el encargado de la limpieza de ese piso.

Por último la unidad máxima, hotel. En hotel tenemos acceso a el conjunto de pisos del hotel y es el encargado de reservar habitaciones, con criterios como: cual habitación tiene la cantidad de piezas requeridas, si el cliente quiere algún piso en particular y si quiere baño privado.

3. Diseño.

Para desarrollar este software se ha tenido en consideración la implementación de las diferentes clases que se procederá a detallar

3.1. Hotel

Hotel
-pisito:ArrayList<Pisos> -arch:String -linea
+leer():boolean +escribir() +obtenerCamasPedidas(int,ArrayList<Persona>):Habitaciones +obtenerCamasPedidas(int):boolean +desalojar(int):boolean +alojar(ArrayList<Persona>,Habitaciones) +verRegistros(int,int):String +obtenerHabitacion(int):Habitaciones

Figura 1: Clase Hotel

Los atributos de la Clase son

- pisito: una lista la cual contiene la cantidad de pisos del hotel
- arch: archivo del cual se genera el hotel, cantidad de pisos, habitaciones y que contienen cada una de ellas.
- linea: es utilizada como variable auxiliar a la hora de leer el archivo (arch)

Los métodos de la clase son:

- leer: Permite sacar la información almacenada en el archivo csv en el cual existe toda la información sobre el hotel
- escribir: Función la cual genera un archivo csv con la información actual del hotel
- obtenerCamasPedidas: este método verifica si existe habitación disponible para una exigencia específica de personas.
- desalojar: deja en estado disponible aquella habitación que haya sido desalojada y limpiada.

- alojar: deja en estado ocupada a la habitación.
- verRegistros: función la cual genera un reporte de toda la información del hotel(cantidad de pisos, habitaciones y la disponibilidad de las mismas).
- obtenerHabitacion retorna la habitación en cuestión pedida por el número de habitación.

3.2. Pisos

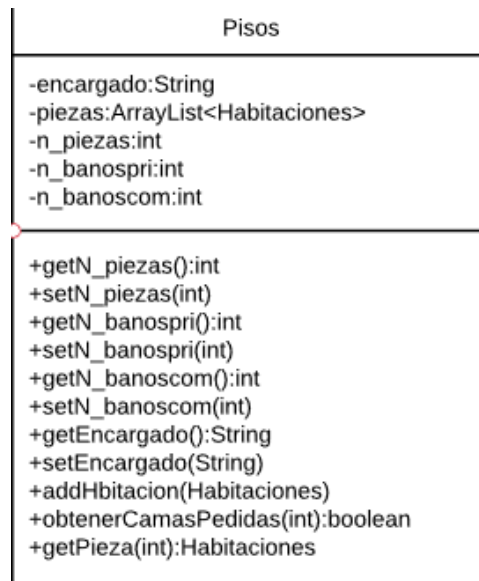


Figura 2: Clase Pisos

Los atributos de la Clase son:

- encargado: Nombre del encargado de limpieza del piso.
- piezas: Lista piezas pertenecientes a un piso.
- npiezas: Cantidad numérica de piezas por piso.
- nbanospri: Cantidad numérica de baños privados(baños en habitaciones).

- nbanoscom: Cantidad numérica de baños comunes.

Los métodos de la clase son:

- getter y setters: Son todos los métodos por los cuales podemos, fuera de la Clase Pisos, obtener y establecer los valores de los atributos de Clase.
- addHabitacion: Es el método por el cual añadimos una habitación al piso en cuestión.
- obtenerCamasPedidas: Recibe la cantidad de camas solicitadas por el cliente y devuelve la disponibilidad que existe de su búsqueda.
- getPieza: Mediante el numero de habitacion, devuelve la habitacion en cuestión.

3.3. Habitaciones



Figura 3: Clase Habitaciones

Los atributos de Clase son

- Nhabitacion: Es el número designado a la habitación en cuestión
- teléfono: Es el método de contacto desde la recepción con la habitación
- Ncamas: Número de camas con que cuenta la habitación
- np: Número de personas que residen actualmente la habitación

- clase: Tipo de la habitacion. Economica, lujosa, etc.
- Huéspedes: Lista de personas que residen en la habitación(distinto al número de ellos)
- monto: Es el precio de la habitacion por día.
- estado: Un número entero que va desde el 0 hasta e 2, el cual describe la disponibilidad de la habitación. (0 = no habilitada; 1 = habilitada y 2 = en limpieza)
- banoprivado: Si tiene o no baño privado.

Los métodos de la clase son:

- getter y setters: Son todos los métodos por los cuales podemos, fuera de la Clase Habitaciones, obtener y establecer los valores de los atributos de Clase.
- obtenerPersonas: Ingresa las personas a las habitaciones.
- getPersona: Este método retorna una persona de una habitacion.
- vaciar: Elimina a todas las personas y deja el estado de la habitacion en 0.
- hospedar: Crea una lista de personas y los ingresa a la habitacion deseada.

3.4. Persona

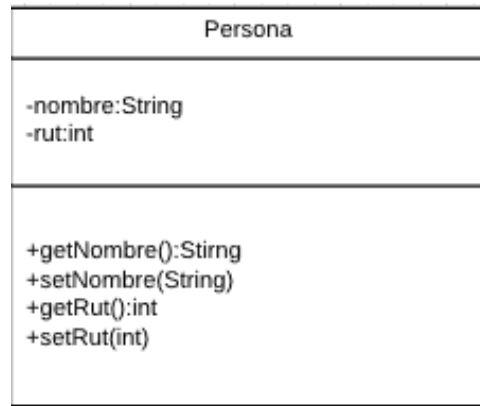


Figura 4: Clase Persona

Los atributos de Clase son:

- nombre: es el nombre de la persona
- rut: es el rut de la persona

Los métodos de la clase son:

- setter y getters: En esta clase los únicos métodos que hay son los que dejan obtener la información de sus atributos por Clases externas.

3.5. interfaces gráficas

las interfaces gráficas interactúan entre si para ingresar a los huéspedes a sus habitaciones correspondiente.



Figura 5: ventana principal

esta ventana es la primera que el usuario ve y es la que tiene como menú las siguientes opciones:

- ingresar huespedes: es la opcion que te dirige a la pantalla donde se registra a las personas
- Mostrar habitaciones: presionando este botón mostrará las habitaciones del hotel con todos sus atributos.
- Actualizar Registro: sirve para subir el archivo csv si existen modificaciones al mismo.

4. Planificación.

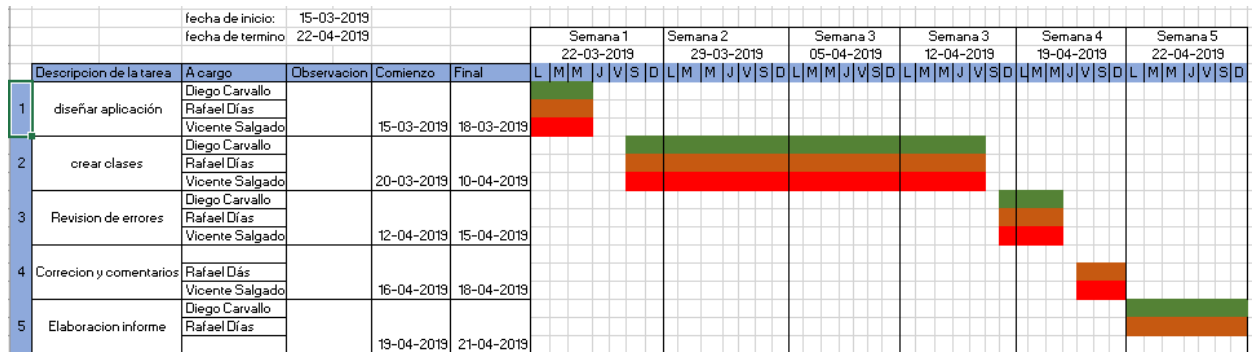


Figura 6: carta gantt equipo

diego.png

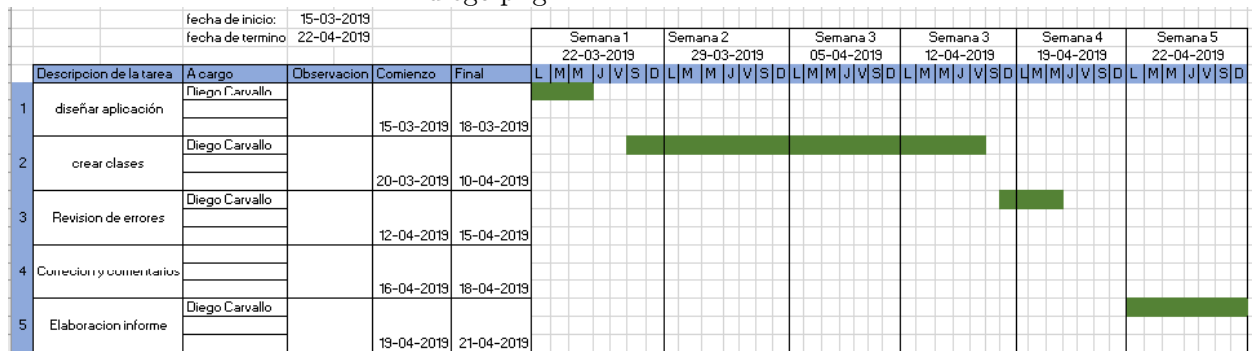


Figura 7: carta gantt Diego Carvallo

rafa.png

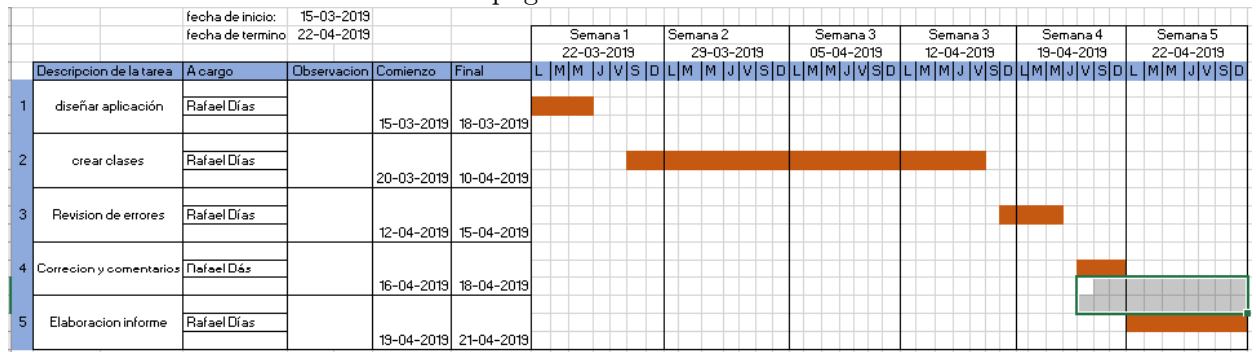


Figura 8: carta gantt Rafael Días

vicho.png

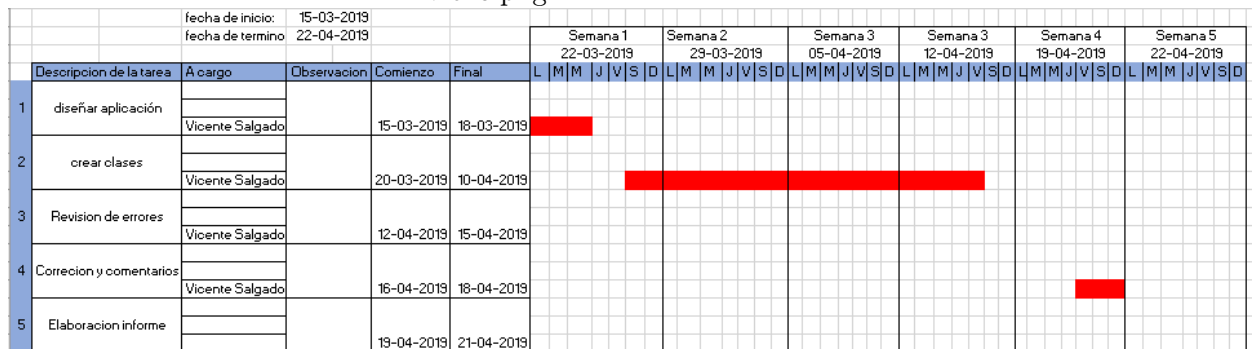


Figura 9: carta gantt Vicente Salgado

5. Conclusiones.

En esta primera entrega, dado los conocimientos adquiridos en el lenguaje Java y la aproximación que se ha obtenido al paradigma de la programación orientada a objetos, es que debemos decir que este avance de proyecto ha sevido más que solo una nota. Ha aportado al entendimiento del comportamiento de un ebjeto y sus diferentes clases, como se modela un porblema, quizás no de la manera máseficiente, pero eso se mejorará con el pasar de las entregas.

No queda más que decir que siendo el primer avance, la aplicación mutará a una mejor versión de si misma a su vez que nosotros mismos como programadores, entrenandonos en elementos fundamentales de la programación como las buenas costumbres.