



**Carrera:** Analista Programador Web

**Materia:** Lenguajes I

**Profesor:** Gauto, David Alejandro

### Trabajo Final

1. Una agencia de viajes mantiene información referida a los hoteles y cabañas que ofrece a sus clientes. La clase Alojamiento se introduce para factorizar atributos y comportamiento. En la aplicación todos los alojamientos son instancias de una de las clases derivadas: Habitación, Hotel o Cabaña (ambas clases heredan de alojamiento). La clase Agencia se implementa como una tabla. La clase cliente tiene acceso a los alojamientos por posición.

Algunas posiciones pueden ser nulas.

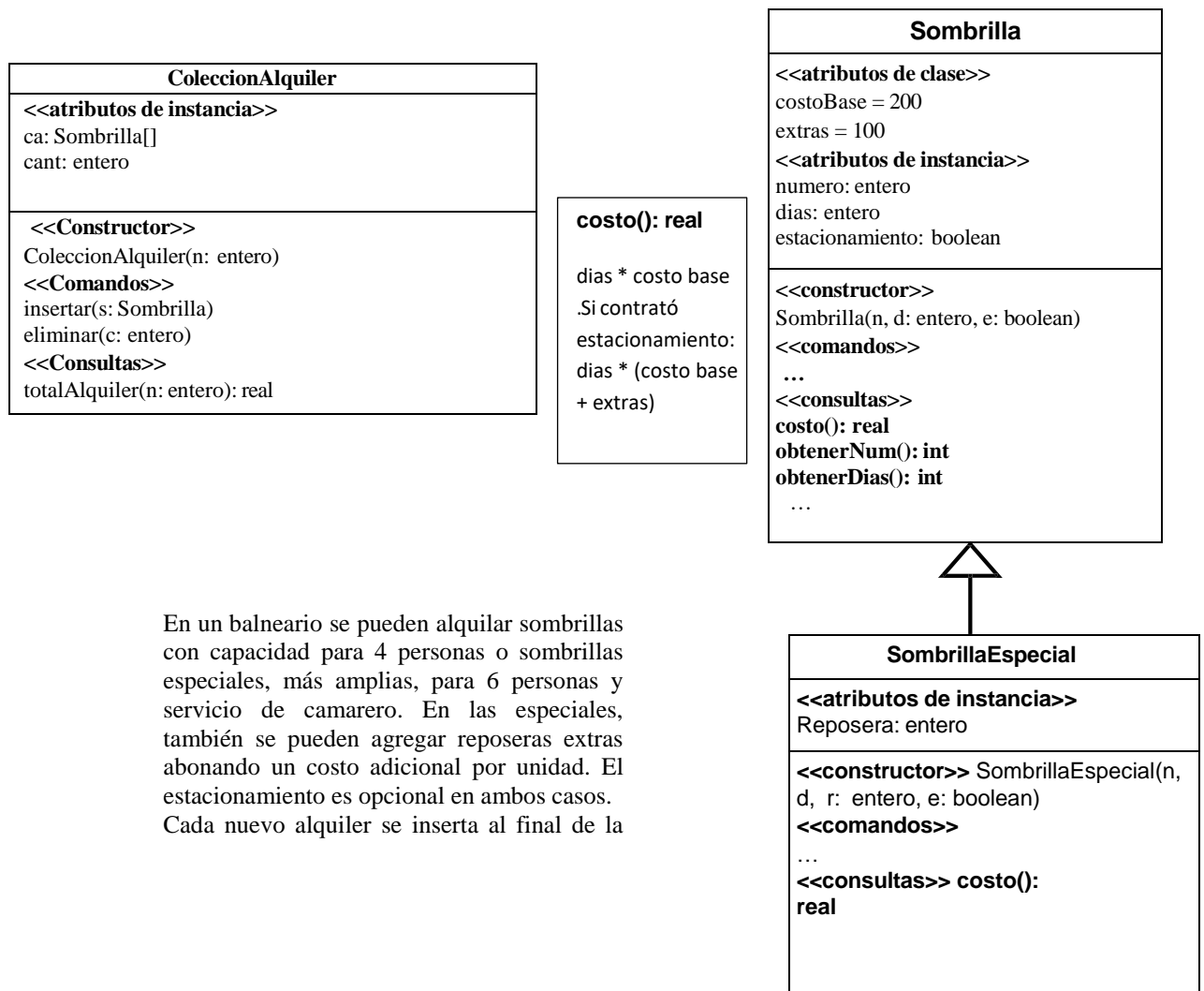
- a) Implemente el diagrama de clases.
- b) Implemente una clase tester que verifique los servicios provistos por la clase Agencia.

<b>Alojamiento*</b> <<atributos de instancia>> codigo: entero domicilio: String TV: boolean <<constructores>> Alojamiento(c: entero, d: String, t: boolean) <<comandos>> establecerDomicilio(d: String) establecerTV (t: boolean) <<consultas>> obtenerCodigo(): entero obtenerDomicilio(): String tieneTV(): boolean toString (): String igualCodigo(a: Alojamiento): boolean estrellas(): entero	<b>Agencia</b> <<atributos de instancia>> alojamientosAgencia: Alojamiento[] <<constructor>> Agencia(n: entero) <<comandos>> insertarAlojamiento(a: Alojamiento, p: entero) eliminarAlojamiento(pos: entero) eliminarAlojamiento(a: Alojamiento) <<consultas>> cantAlojamientos(): entero recuperarAlojamiento(pos:entero): Alojamiento recuperarPosicion(a: Alojamiento): entero estaAlojamiento(c: entero): Alojamiento estaLlena(): boolean hayAlojamientos(): boolean masEstrellas(cantEst: entero): Agencia
HabitacionHotel	Cabaña

<<atributos de instancia>> precioXPersona: real piscina: boolean	<<atributos de instancia>> precioXDia: real cantHabitaciones: entero cantBanios: entero
<<constructores>> HabitacionHotel(c:entero,d:String,t,p:boolean, r:real) <<comandos>> establecerPrecioXPersona(r: real) establecerPiscina(p: boolean) <<consultas>> obtenerPrecioXPersona(): real tienePiscina(): boolean estrellas(): entero	<<constructores>> Cabaña(c:entero,d:String,t:boolean,p:real, ch,cb: entero) <<comandos>> establecerPrecioXDia(r: real) <<consultas>> obtenerPrecioXDia(): real

- **estrellas(): entero** en Alojamiento retorna 2 si el alojamiento tiene TV, 1 en caso contrario.
- **estrellas(): entero** en HabitacionHotel retorna la cantidad de estrellas de un Alojamiento más 1 si tiene piscina.
- **recuperarPosicion(a: Alojamiento) : entero** busca por identidad.
- **estaAlojamiento(c: entero): Alojamiento** busca por código.
- **masEstrellas(cantEst: entero): Agencia** en Agencia retorna una nueva agencia sólo con los alojamientos que tengan más de cantEst estrellas.
- **hayAlojamientos(): boolean** en Agencia retorna true si y sólo si en la tabla hay algún alojamiento.

2.



colección incrementando la cantidad. **La colección NO está ordenada.** Si se almacenan  $n$  alquileres, se ocupan las primeras  $n$  componentes del arreglo.

***insertar(s: Sombrilla)*** agrega la sombrilla pasada por parámetro a la colección. Requiere haber controlado que existe lugar disponible. ***eliminar(c: entero)*** elimina de la colección el objeto correspondiente al número de sombrilla pasado por parámetro. La colección debe quedar compactada nuevamente.

***totalAlquiler(n: entero): real***  
devuelve el monto total que se obtendrá por las sombrillas que superen los  $n$  días de alquiler.

**Implementar las tres clases indicadas en el diagrama.**

**costo(): real**

140% del costo de la sombrilla más (1/3\*extras)  
por reposera por día