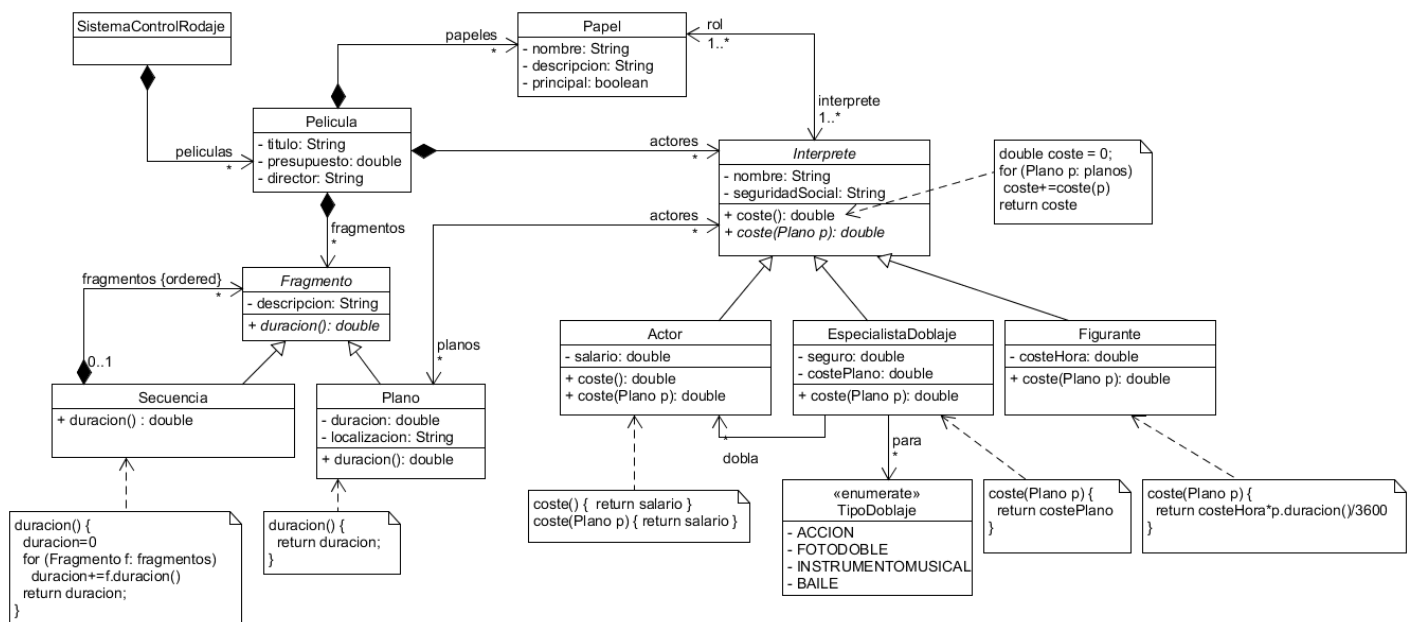


Entrega cada ejercicio en hojas separadas



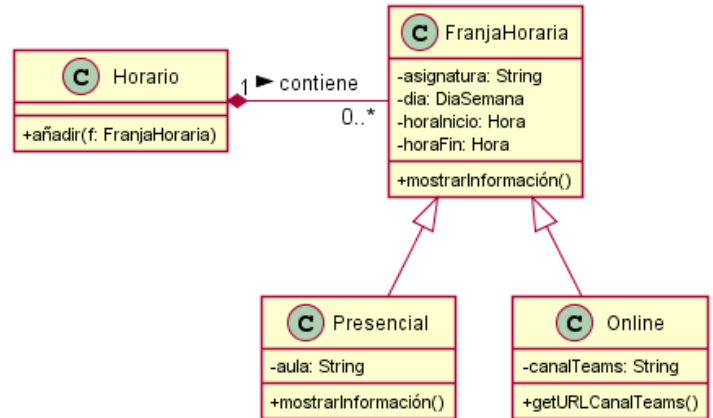
Prueba 1 de Evaluación Continua

Análisis y Diseño de Software (2020/2021)

Entrega cada ejercicio en hojas separadas

Ejercicio 2 (2.5 puntos)

El diagrama de clases de la figura muestra un diseño para manejar horarios de clases, donde estas pueden ser presenciales u online. Analízalo y contesta a las siguientes cuestiones:



1. (0.5 ptos) Para cada una de las siguientes líneas de código, indica si son correctas o incorrectas y señala por qué.

- a) (0.25 ptos) `FranjaHoraria c1 = new Presencial();`
- b) (0.25 ptos) `Presencial c2 = new FranjaHoraria();`

Solución:

- a) CORRECTA: se instancia un objeto de clase “Presencial” y se le asigna a una referencia a un objeto de clase FranjaHoraria (clase padre).
- b) INCORRECTA: un constructor de la clase padre no puede emplearse para instanciar un objeto de las clases hijas. Cualquier “FranjaHoraria” no puede ser “Presencial”.

2. (1.0 ptos) Partiendo del siguiente fragmento de código Java para instanciar objetos de diferentes tipos de clases:

```
FranjaHoraria c1 = new FranjaHoraria();
Online c2 = Online();
Presencial c3 = Presencial();
```

Para cada una de las siguientes asignaciones, indica si es correcta o incorrecta y por qué.

- a) (0.25 ptos) `c2 = c1;` // INCORRECTO: Cualquier FranjaHoraria no puede considerarse Online
 - b) (0.25 ptos) `c1 = c2;` // CORRECTO: Cualquier Online es una FranjaHoraria
 - c) (0.25 ptos) `c1 = c3;` // CORRECTO: Cualquier Presencial es una FranjaHoraria
 - d) (0.25 ptos) `c2 = c3;` // INCORRECTO: Una Online no puede ser Presencial
3. (0.25 ptos) Teniendo en cuenta que Java emplea ligadura dinámica (o linkado dinámico), ¿a qué clase (FranjaHoraria, Presencial u Online) pertenece el método que se ejecuta en el siguiente fragmento de código y por qué?

```
Presencial c1 = new Presencial();
FranjaHoraria c2 = c1;
c2.mostrarInformación();
```

Solución: Se ejecuta el método `mostrarInformación()` de la clase “Presencial” porque el linkado dinámico hace que al asignar `c1` a `c2`, la referencia `c2` apunte al área de memoria reservada para el objeto de clase “Presencial”.

4. (0.25 pts) Indica qué cambiaría en el diagrama anterior para limitar Horario solo tenga instancias de franjas horarias que sean “Presencial” u “Online” pero nunca “FranjaHoraria” sin indicar ninguno de sus subtipos.

Solución: Basta con hacer que `FranjaHoraria` sea abstracta, obligando así a que no se puedan crear directamente instancias de “FranjaHoraria”.

5. (0,5 pts) Señala si es posible invocar los métodos en cada uno de los siguientes fragmentos y por qué.

- a) (0,25 pts)

```
FranjaHoraria c1 = new Online();  
c1.getURLCanalTeams();
```

- b) (0,25 pts)

```
Online c1 = new Online();  
c1.mostrarInformación();
```

Solución:

- a) No es posible porque la referencia `c1` es de tipo `FranjaHoraria`.
b) Es posible porque se está empleando la implementación de la clase padre.

Prueba 1 de Evaluación Continua

Análisis y Diseño de Software (2020/2021)

Entrega cada ejercicio en hojas separadas

Ejercicio 3 (3 puntos)

Se quiere desarrollar una aplicación que gestione las reservas de espacios y materiales en la EPS. El sistema gestionará y ofrecerá información sobre usuarios, ítems reservables (espacios y materiales) y reservas realizadas.

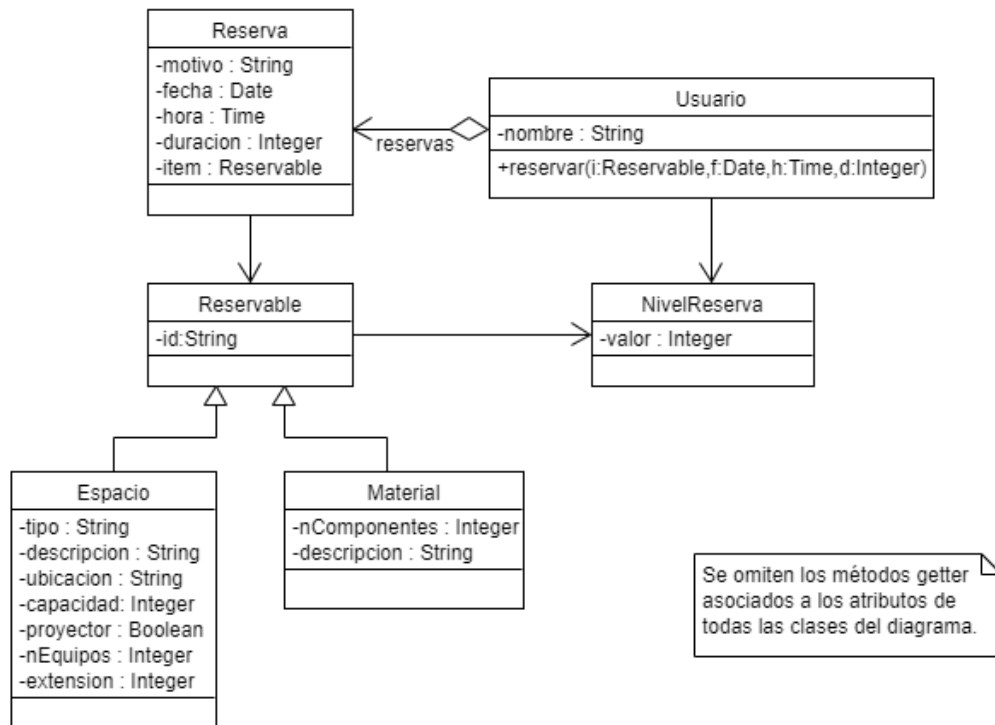
Los usuarios han de solicitar al sistema las reserva de ítems. Los espacios son de diversos tipos: aulas (con o sin proyector), laboratorios (con cierto número de equipos informáticos) y salas de reuniones (con una extensión telefónica).

Una reserva se podrá realizar siempre que el nivel de reserva (un número entero) del usuario sea igual o inferior al nivel de reserva del ítem. Se solicitará para una fecha y hora de inicio y para una fecha y hora de fin, y se facilitará la obtención de su duración en minutos.

Atendiendo a estos requisitos, se ha realizado el diagrama de clases de abajo.

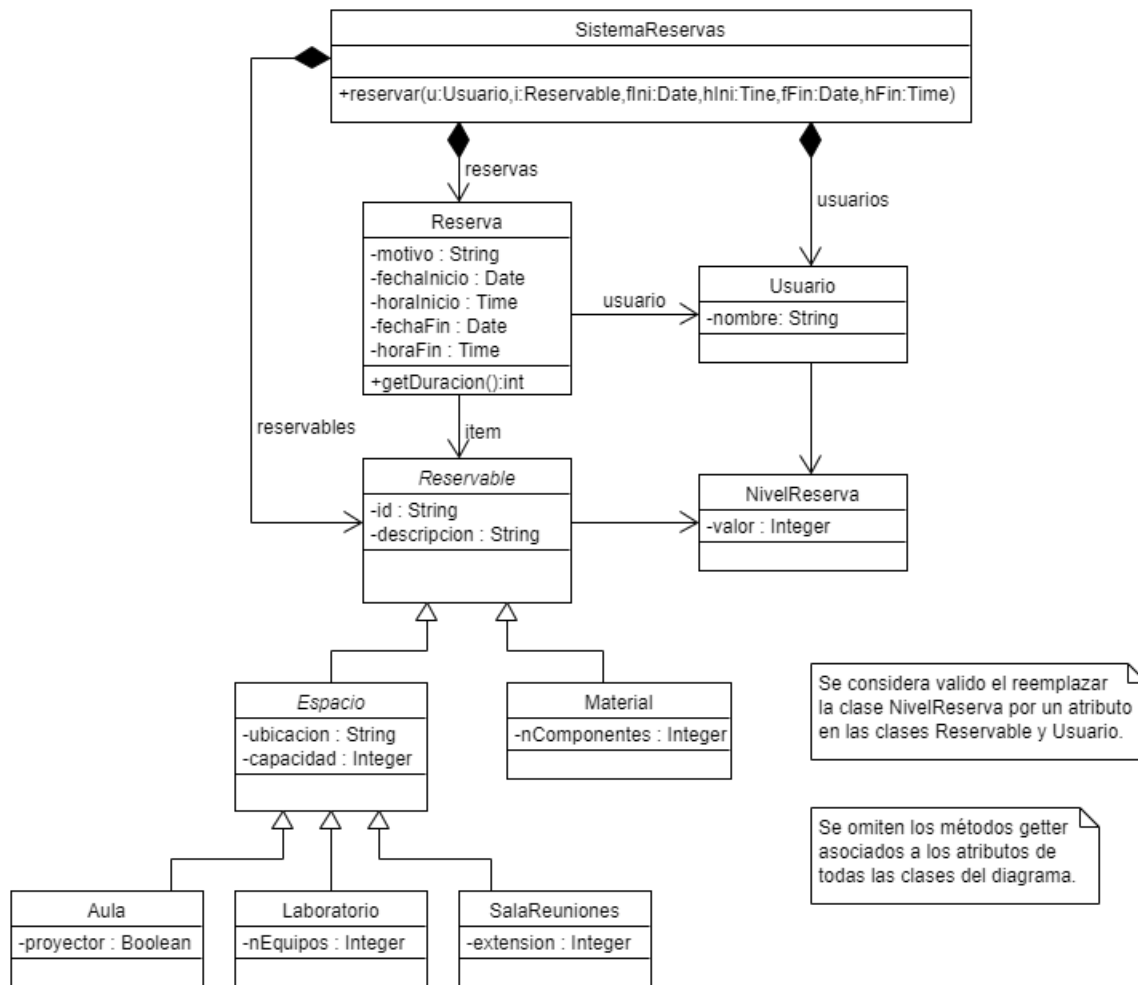
Se pide:

Modificar y completar si es necesario el diagrama para que refleje un buen diseño orientado a objetos. Entrega un nuevo diagrama que refleje el nuevo diseño.



Solución:

El siguiente diagrama podría ser una solución al problema planteado:



Los cambios realizados son los siguientes:

1. Clase `SistemaReservas`: se añade al diagrama.
2. Método `reservar`: se modifica su declaración.
3. Método `reservar`: se sitúa en la clase `SistemaReservas`.
4. Agregaciones fuertes de la clase `SistemaReservas`: se añaden al diagrama.
5. Agregación débil de la clase `Usuario`: se elimina, sustituyéndola por relaciones de asociación entre `Usuario` y `Reserva`.
6. Clases `Espacio` y `Reservables`: se declaran como abstractas.
7. Clases `Aula`, `Laboratorio`, `SalaReuniones`: se añaden como subclases de `Espacio`.
8. Clase `Reserva`: en ella se añaden atributos `fechaFin` y `horaFin`, y se añade el método `getDuracion`.
9. Clase `Reserva`: en ella se elimina al atributo `item` por ser redundante con la relación entre las clases `Reserva` y `Reservable`.
10. Clase `Reservable`: en ella se coloca el atributo `descripcion`, común a sus subclases `Espacio` y `Material`, de las que se elimina.