 UNIVERSIDAD DE GRANADA Dpto. LSI	Programación y Diseño Orientado a Objetos (PDOO) Grado en Ingeniería Informática Evaluación Continua: Primer Parcial de Prácticas (P1, P2 y P3)
--	--

Apellidos:

Nombre: Grupo:

INSTRUCCIONES:

- Escriba en cada folio «APELLIDOS, NOMBRE» (en MAYÚSCULAS) y entregar al final.
- Acceda a su cuenta personal especificando el código **w10pdooviex**.
- Acceda a Prado para descargar ficheros adicionales de esta prueba a través del enlace a Chrome en el escritorio.
- Acceda desde el escritorio a la unidad de disco «Home» y guarde allí sus ejercicios del examen, ya que, si el ordenador se reinicia o se cuelga, al volver a arrancar la imagen seguirán estando los ficheros ahí almacenados.
- Entregue un único archivo zip con el proyecto en Java completo (en Netbeans, opción exportar proyecto) y capturas de pantalla **completa** de la salida de la ejecución.

Consideraciones iniciales

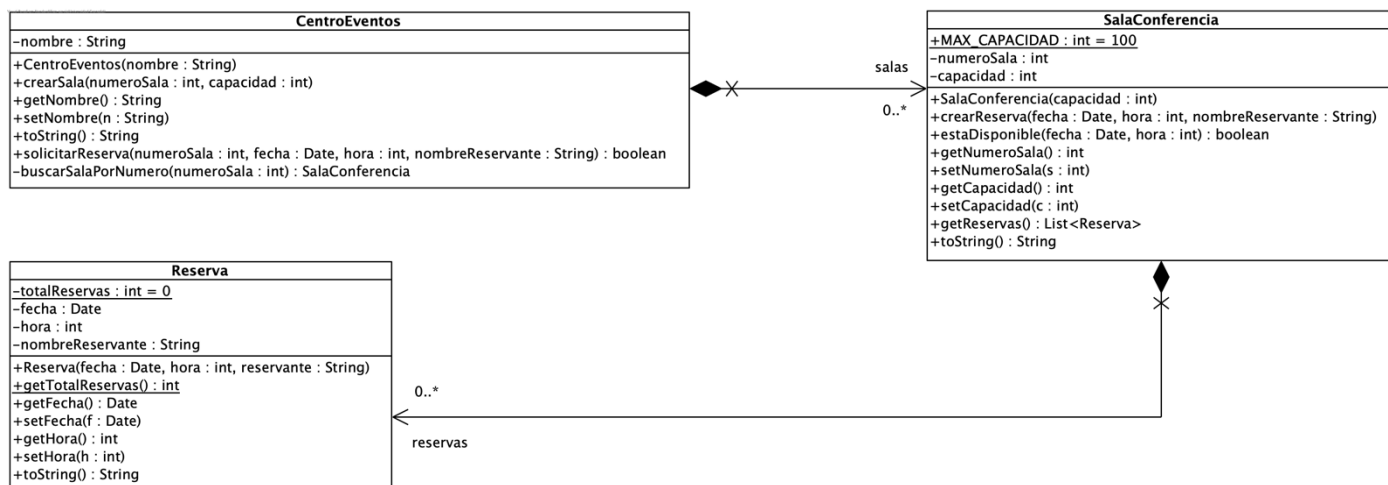
Consideración 1. [0,5 puntos] Asegure un diseño coherente y bien estructurado de su código en todo momento, como la generación de **3 tipos de constructores mínimo** (con todos los atributos, uno por defecto y uno de copia) —y no repita código innecesariamente (reutilización)—. *Cuidado con la composición, asegúrese de no comprometer la privacidad de la creación de instancias de las PARTRES desde fuera del TODO.* Además, todos los **getters/setters** deseados ya se han indicado en el diagrama de clases (*si necesita alguno más, adelante; pero, cuidado con comprometer la privacidad*).

Consideración 2. [0,25 puntos] Evite usar múltiples retornos en sus métodos optando, en cambio, por un único punto de retorno por razones de diseño.

Consideración 3. [0,25 puntos] Evite el uso de «números mágicos» en su código. Además, en lugar de asignar valores booleanos directamente como true o false, construya expresiones booleanas a partir de las variables y condiciones pertinentes.

Descripción de la actividad

Se desea implementar una aplicación para la gestión de una biblioteca de música digital según el siguiente diagrama UML:



Ejercicio 1. [1,5 puntos] Implemente la clase **Reserva** según se indica en el diagrama de clases y en las consideraciones previas. Tenga en cuenta que queremos saber el número de reservas totales del sistema.

Ejercicio 2. [1,5 puntos] Implemente la clase **SalaConferencia** según se indica en el diagrama de clases y en las consideraciones previas. Asegure las restricciones de máxima capacidad en la creación de una sala (y cuidado con repetir código; reutilice siempre).

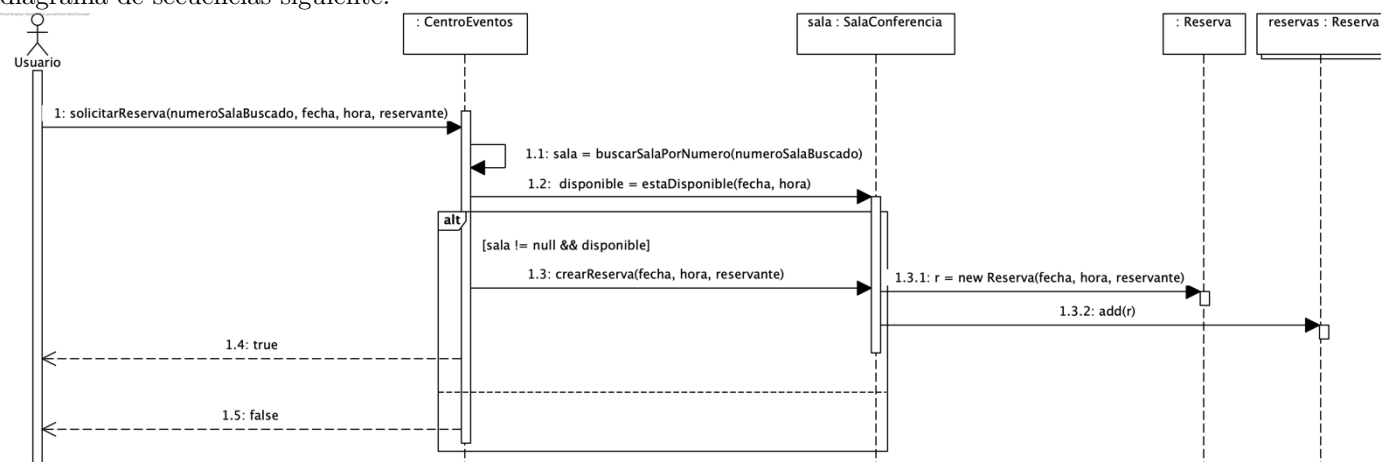
Ejercicio 3. [0,5 puntos] En **SalaConferencia**, el método *estaDisponible* se utilizará para verificar si ya existe una reserva para la fecha y hora solicitadas (dicho de otra manera, comprobar todas las reservas para dicha sala y ver si la fecha y hora no son iguales). Recuerda que *Date* tiene disponible el método *equals*; y los tipos primitivos como *int* su *equals* es *==*, para comprobar igualdad.

Ejercicio 4. [0,5 puntos] En **SalaConferencia**, el método *crearReserva* tiene que comprobar si la sala está disponible para la fecha-hora indicadas. En caso afirmativo, creará una nueva reserva y la añadirá a la lista de *reservas*.

Ejercicio 5. [2 puntos] Implemente la clase **CentroEventos** según se indica en el diagrama de clases y en las consideraciones previas.

Ejercicio 6. [1 punto] En **CentroEventos**, para implementar el método *buscarSalaPorNumero*, debe recorrer las salas que tiene el centro, y si encuentra el número de la sala que se busca, devolverá esa sala; sino devolverá *null*. Atención, este método no tiene que comprobar si la sala está disponible.

Ejercicio 7. [1 punto] En **CentroEventos**, para implementar el método *solicitarReserva*, siga las indicaciones del diagrama de secuencias siguiente:



Ejercicio 8. [1 punto] En el programa principal, pruebe lo siguiente:

- [0,05 puntos]** Cree un nuevo centro de eventos con el nombre «Centro de Convenciones Internacional».
- [0,25 puntos]** Al centro que acabas de crear, dale la orden de crear una nueva SalaConferencia cuyo número de sala es 101 y la capacidad es de 50. Dale la orden ahora de crear otra sala con el número 102 y capacidad 200. Imprime por consola en ambos casos el tamaño de las salas y verifica que al final solo deberías tener 1 sala.
- [0,25 puntos]** Tal y como se presenta en el diagrama de secuencias, llame al método *solicitarReserva* adecuadamente para realizar una reserva. Imprima por consola si la reserva se realiza o no (debe ser exitosa).
- [0,25 puntos]** Ahora, intente solicitar una nueva reserva en las mismas condiciones que la reserva anterior, es decir, de la misma sala, fecha y hora. Imprima por consola si la reserva se realiza o no (debe no ser exitosa).
- [0,20 puntos]** Por último, imprima el número total de reservas del sistema, ya que disponemos de una lógica para ello.
- [1 punto para M.H.]** Crea un nuevo centro a partir del anterior. Quita todas las reservas preexistentes. Añádale al menos una reserva nueva a este centro que sea correcta e imprime por consola el centro.