|  |  |  |
| --- | --- | --- |
| C:\Users\afreijo\Documents\uad.jpg | **UNIVERSIDAD ARGENTINA DE LA EMPRESA** | |
| **Departamento de Tecnología**  **Informática**  PARADIGMA ORIENTADO A OBJETOS (3.4.208)  Profesor: Ignacio Colombo | |
| Examen Recuperatorio  ALUMNO: MIGUEL ZAMBRANO LU: 1174581 FECHA: 04/12/2024  CARRERA: ING. INFORMATICA  TEMA: Recuperatorio | | |
| NOTA: EL EXAMEN ESCRITO ES UN DOCUMENTO DE GRAN IMPORTANCIA PARA LA EVALUACIÓN DE LOS CONOCIMIENTOS ADQUIRIDOS, POR LO TANTO, SE SOLICITA LEER ATENTAMENTE LO SIGUIENTE:   1. El parcial se entregará subiendo los archivos en un Pull Request en el repositorio GitHub del alumno.    * El repositorio debe ser el mismo que el alumno utiliza para los ejercicios en clase. 2. No se aceptara ningún commit con horario posterior a la finalización del examen.    * En caso de no poder recuperar el commit con horario dentro del examen, significara la desaprobación del mismo. 3. Los diagramas a entregar son en formato PNG. En adición, se debe entregar el proyecto (si se desarrolló con startUML) o el código en un archivo .txt (si se desarrolló en PlantUML). 4. El link al Pull Request debe entregarse en la tarea de Teams correspondiente al examen. 5. El criterio de evaluación es del 50% del examen correcto, esto incluye desde la claridad de los nombres de las clases, métodos y atributos, su aplicación al estándar UML (asociaciones, cardinalidad, etc.) y aplicación de los conceptos y buenas prácticas de la materia. **Revisar la rúbrica de evaluación adjunta al examen.**    * Para llegar al 50%, debe tener al menos:      1. Bien el diagrama de clases,      2. Bien en al menos uno de los diagramas de secuencia, **y**      3. Resolver bien la problemática planteada utilizando los conceptos de la materia. 6. El tiempo estipulado de desarrollo del examen es de 2 (dos) horas.   Por último, les recuerdo que es una instancia de evaluación individual, así que el intercambio de información entre los alumnos se considerará copia con las sanciones del caso si correspondiere. | |  |

## Descripción de la situación

Se desea desarrollar un sistema de gestión para la reserva de canchas de deporte, incluyendo fútbol, pádel y tenis.

## Detalles del negocio

El sistema debe permitir a los usuarios reservar canchas para diferentes deportes en distintos horarios. Cada cancha tiene un código único de identificación y está asociada con un tipo de deporte (fútbol, pádel, tenis). Las reservas pueden ser realizadas por usuarios registrados, quienes deben cumplir con ciertas restricciones para poder realizar una reserva.

Tipos de Canchas:

* Fútbol: Canchas de césped sintético y natural.
* Pádel: Canchas cubiertas y descubiertas.
* Tenis: Canchas de arcilla y cemento.

Restricciones para Reservas:

* Un usuario no puede tener más de dos reservas activas al mismo tiempo.
* No se puede reservar una cancha si ya está reservada en el mismo horario.

Costo de Reserva: El costo de la reserva varía según el tipo de cancha y la duración de la reserva. La fórmula para calcular el costo es la siguiente:

|  |  |
| --- | --- |
| **Tipo de Cancha** | **Fórmula de Costo** |
| Fútbol | Precio base \* cantidad de jugadores |
| Pádel | Precio base + precio alquiler de paleta + precio alquiler de pelotas |
| Tenis | Precio base + precio alquiler de raqueta |

## Puntos Para Desarrollar

1. Desarrollar el Diagrama de Clases
2. Desarrollar los diagramas de Secuencia de los casos de uso:
   1. Registrar una Reserva, que recibe por parámetro:
      1. idClub
      2. numeroCancha
      3. idCliente
      4. fecha
      5. hora

**Tener en cuenta las restricciones correspondientes.**

* 1. Calcular el Costo de una Reserva (previamente registrada), que recibe por parámetro el idReserva y retorna el valor en pesos del costo.

## Rúbrica de Evaluación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | NIVELES DE LOGRO | | | |
| **MUY BIEN (2,5 puntos)** | **BIEN (1,5 puntos)** | **REGULAR (0,5 puntos)** | **NO SATISFACTORIO (0 puntos)** |
| **Diagrama de Clases** | Diagrama completo incluyendo todas las clases y relaciones con sus tipos (agregación, composición). Todos los tipos de relaciones son correctos. Se definen correctamente los atributos con su accesibilidad y tipo. Los métodos contienen la accesibilidad, parámetros y tipo de retorno. | Diagrama incluye todas las clases y relaciones. Pocos tipos de relaciones son omitidos o incorrectos. Se definen correctamente atributos (con accesibilidad y tipo). Se definen pocos métodos pero contienen accesibilidad, parámetros y tipo de retorno. | Diagrama incluye todas las clases. Pocos tipos de relaciones son omitidos o incorrectos. Se omiten algunos atributos o métodos claves. Algunos métodos o atributos no contienen la accesibilidad o tipo correcto. | Diagrama omite clases y/o relaciones. No presenta ningún tipo de relación o en su mayoría son incorrectos. Omite atributos y métodos claves. Define incorrectamente accesibilidad, tipos y parámetros. |
| **Diagrama de Secuencia (x2)** | Diagrama completo y con un buen nivel de detalle de los métodos. Los mensajes cumplen con el estándar (nombre, tipo de retorno, parámetros). Se respeta la secuencialidad del diagrama. Los elementos del diagrama (loop, alt, opt, etc.) están bien ubicados y siguen el estándar UML. | Diagrama completo pero omite detalle de algunos métodos. Los mensajes cumplen con el estándar. Se respeta la secuencialidad. Algunos elementos del diagrama pueden no estar bien ubicados. | Diagrama no está completo en su totalidad para resolver la funcionalidad u omite detalle de varios métodos. Las clases utilizadas son correctas y la secuencialidad se respeta. Algunos mensajes no respetan el estándar o los retornos son omitidos. Algunos elementos del diagrama pueden no estar bien ubicados. | Diagrama incompleto o no detalla ningún método. No respeta la secuencialidad. No respeta el estándar UML del diagrama. Tipos de retorno son omitidos o no utilizan la flecha correspondiente. Varios elementos del diagrama son omitidos o están mal ubicados. |
| **Resolución de la problemática** | La problemática planteada en el enunciado es resuelta en los diagramas presentados. La resolución aplica los conceptos del paradigma vistos en la materia y las buenas prácticas. | Codigo cumple con los patrones de diseño MVC y las buenas practicas. Se utiliza ObjectView para enviar datos a la pantalla. Alguno de los requerimientos del código es resuelto. | Codigo cumple con los patrones de diseño MVC y las buenas practicas. Se utiliza ObjectView para enviar datos a la pantalla. Más de un requerimiento no se cumple. | No respetan las buenas prácticas de MVC. |