|  |  |  |
| --- | --- | --- |
| 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 Parcial  ALUMNO: DAMM NICOLAS LU:1186157 FECHA: 23/10/2024  CARRERA: INGENIERIA EN INFORMATICA TEMA: 1 | | |
| 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 un parque natural.

## Detalles del negocio

El parque natural está dividido en varias áreas protegidas, cada una con su propia flora y fauna. Cada área está identificada por un código único y tiene una ubicación y una capacidad máxima de visitantes permitidos por día.

Los visitantes pueden registrarse para visitar el parque y deben comprar un boleto que les permite acceder a una o más áreas protegidas. Los boletos tienen un costo que varía según la cantidad de áreas que el visitante desea visitar, la temporada de la visita y el tipo de boleto que se adquiere. Los tipos de boletos pueden ser estudiantiles, de temporada baja o de temporada alta. Cada boleto sirve para un solo día y su precio se calcula en base a las siguientes formulas:

|  |  |
| --- | --- |
| **Tipo Boleto** | **Formula de Precio** |
| Estudiantil | 25% descuento sobre el precio de temporada baja (sin importar para que temporada solicita) |
| Temporada Baja | precio base \* cantidad de áreas |
| Temporada Alta | precio base \* cantidad de áreas \* 3 |

Los boletos, una vez adquiridos, son Emitidos. Si el visitante accedió al parque el día correspondiente, el boleto debe marcarse como utilizado, pero si pasa el día y el visitante no se presenta, el boleto se marcara como vencido.

El sistema debe permitir a los administradores del parque:

* Registrar nuevas áreas protegidas.
* Registrar visitantes y emitir boletos.
* Controlar la capacidad de visitantes en cada área.
* Generar informes sobre la cantidad de visitantes y el estado de las áreas protegidas.

**IMPORTANTE:** Un boleto no se puede emitir si para la fecha solicitada se sobrepasa la cantidad permitida en alguna de las áreas solicitadas. Por ejemplo, si para el área Norte la capacidad es de 20 personas y ya hay emitidos 20 boletos con fecha del 23/10 que incluyen esa área, no se puede permitir emitir ningún boleto más para ese día en esa área del parque.

## Puntos Para Desarrollar

1. Desarrollar el Diagrama de Clases
2. Desarrollar los diagramas de Secuencia de los casos de uso:
   1. Emitir un boleto estudiantil que recibe por parámetro el DNI del visitante, fecha de visita y la lista de IDs de las áreas a visitar.
   2. Calcular el precio final de un boleto (previamente registrado), del cual se recibe el identificador del mismo.

## 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. | La problemática general es resuelta pero falta cierto nivel de detalle o contemplar alguna condición particular. Se aplican los conceptos del paradigma y las buenas prácticas. | La resolución omite algunas condiciones presentadas en la problemática. Se aplican los conceptos del paradigma y las buenas prácticas. | No se resuelve la problemática o se omiten varias condiciones de ella. No se aplican los conceptos claves del paradigma y las buenas prácticas. |