

1° Parcial Lab II - 2°A - 2°Cuat. 2022

Requerimientos

Nos han contratado para desarrollar el sistema que facilite la administración de los servicios que ofrece una compañía de **CRUCEROS**. La misma cuenta con una flota que permite viajar a distintos destinos regionales y extra regionales.

Los cuatro vendedores de viajes designados por la empresa deberán poder registrarse cada uno de ellos con su usuario y contraseña para luego poder realizar las siguientes acciones:

1. Visualizar la lista de viajes disponibles y la información del mismo.

Deberá visualizarse en sistema todos los viajes disponibles y poder crear nuevos viajes. Una vez elegido el crucero, deberá poder visualizarse la información básica del viaje, tales como el origen, el destino, hora de partida, precio, y alguna información más que crean conveniente.

2. Vender un crucero a un pasajero o a un grupo familiar.

Deberá poder vender pasajes, siempre y cuando se pueda satisfacer las necesidades del cliente (por ejemplo, si ofrece o no gimnasio, si tiene piscina, si posee el suficiente espacio, etc) y si cumple con validaciones básicas:

Si el crucero está lleno, o si su bodega no tiene más capacidad para recibir las valijas de los pasajeros, si el crucero está en viaje o ya regresó.

3. Información de cada viaje.

El vendedor registrado deberá acceder a ver la información detallada de los pasajeros.

4. Consultar estadísticas históricas.

En alguna sección, deberán poder verse los viajes que ya han sido realizados, para poder efectuar cálculos estadísticos sobre ellos, por ejemplo: recaudación, cantidad de pasajeros de cada crucero, destino más elegido, etc.

Características de la compañía

A nivel regional (sudamérica), ofrece los siguientes destinos:

- Montevideo (Uruguay)
- Ushuaia (Argentina)
- Recife (Brasil)
- Santiago (Chile)
- Lima (Perú)
- Isla de Pascua (Chile)
- Isla Galápagos (Ecuador)
- Puerto Madryn (Argentina)
- Río de Janeiro (Brasil)
- Cartagena (Colombia)

A nivel extra regional (resto del mundo), ofrece solo los siguientes destinos:

- La Habana (Cuba)
- Venecia (Italia)
- Acapulco (México)
- Miami (EE.UU.)
- Nueva York (EE.UU.)
- Bangkok (Tailandia)
- Taipei (Taiwán)
- Atenas (Grecia)

Cada una de las 7 embarcaciones poseen (al menos) las siguientes características:

- ☐ Se identifica a partir de su matrícula (un identificador alfanumérico de 8 dígitos).
- ☐ Nombre
- ☐ Cantidad de camarotes (con capacidad para 4 personas cada uno).
- ☐ Cantidad de salones (al menos uno, el salón comedor).
- ☐ Cantidad de casinos (puede no tener).
- ☐ Capacidad bodega (en kilogramos).

Por otro lado, cada viaje deberá informar (al menos):

- ☐ Ciudad de partida.
- ☐ Ciudad de destino.
- ☐ Fecha de inicio del viaje.
- ☐ Crucero en el que se viajará.
- ☐ Cantidad de camarotes disponibles para clase premium.
- ☐ Cantidad de camarotes disponibles para clase turista.
- ☐ Costo de cada clase.
- ☐ Duración del viaje (en horas).

De cada crucero, el 35% de sus camarotes serán para la clase premium.

La duración del viaje será calculada de forma aleatoria entre la duración mínima y máxima de cada destino.

Viajes regionales: entre 72 hs. y 360 hs.

Viajes extra regionales: entre 480 hs. y 720 hs.

El costo del pasaje se calculará en base a la duración del mismo y el tipo de clase que se ha elegido:

- ☐ Cada hora de viaje costará 57 dólares en destinos regionales y 120 en destinos extra regionales (Ambos en clase Turista).
- ☐ Clase Turista: Podrá llevar un bolso de mano y despachar una valija de hasta 25 kg.

- ☐ Clase Premium: Podrá llevar un bolso de mano y despachar hasta dos valijas de 25 kg cada una. Esta clase cuesta un 20% más de lo que costaría el mismo pasaje para turista.

De los pasajeros se necesita saber (al menos):

- ☐ Apellidos y nombres completos
- ☐ Pasaporte (con datos completos, no solo el número).
- ☐ Edad.
- ☐ Equipajes con los que viaja (puede ser de mano y/o de bodega).

Consigna

Desarrollar una aplicación de escritorio con **Windows Forms** que resuelva las siguientes necesidades del cliente. Todas las entidades deben realizarse en un proyecto de tipo Librería de clases.

Se deberá poder:

- ☐ Ver la lista de viajes y la cantidad de camarotes disponibles.
- ☐ Ver la lista de pasajeros (con información completa) de cada crucero.
- ☐ Visualizar o consultar por: pasaporte, nombres, apellidos y la edad de los clientes.
- ☐ Contar con una barra de información de la aplicación donde figure el nombre del operador conectado y la fecha actual (sin la hora).
- ☐ Visualizar fácilmente el estado actual del crucero (disponible/completo/etc.).
- ☐ Poder venderle un pasaje a un pasajero y asociarlo a un crucero.
- ☐ De los cruceros se deberá poder ver toda su información bien detallada.
- ☐ Se le deberá mostrar al usuario el costo del viaje, el costo final bruto por el servicio y el neto a pagar una vez aplicado IVA y las tasas correspondientes.
- ☐ Poder visualizar estadísticas históricas:
 - ☐ Lista de destinos ordenados por facturación de forma descendente.
 - ☐ Lista de pasajeros frecuentes ordenadas por cantidad de viajes.
 - ☐ Ganancias totales y clasificadas por servicio (regional / extra regional).
 - ☐ Horas de viaje de cada crucero.
 - ☐ El destino más pedido por los clientes.
 - ☐ etc.

Criterios de evaluación

Formato de entrega

El examen será entregado el día miércoles 05 de Octubre.

Por medio de un formulario de Google se pedirá que registren el repositorio de **github** donde se encontrará ubicado su código.

El mismo debe configurarse como **privado** y agregar los usuarios de los docentes para poder ver el código. Esto se realiza con el fin de que su código no sea 'copiado' sin su consentimiento.

Recordar que, códigos parecidos / muy parecidos / idénticos, implica la reprobación de ambos parciales.

Documentación

Se deberá completar la información indicada en el archivo **README.md** que se encuentra en el repositorio.

Para trabajar con este archivo se deberá utilizar el lenguaje de marcado Markdown.

No se corregirá ni revisará ningún parcial que no presente esta documentación completa.

Secciones e información a documentar

- ☐ **Título:** Ponerle un nombre a la aplicación.
- ☐ **Sobre mí:** Presentarse brevemente. Contar su experiencia programando y lo que significó para vos este trabajo (¿fue un desafío? ¿fue fácil? ¿aprendiste? ¿te divertiste? etc.).
- ☐ **Resumen:** Explicar qué hace la aplicación y cómo se usa a grandes rasgos.
- ☐ **Diagrama de clases:** Pegar una foto del diagrama de clases correspondiente a la lógica de negocio. Se debe construir con la herramienta de [Visual Studio](#) y deberá estar actualizado a la última versión entregada de la solución.
- ☐ **Justificación técnica:** Indicar tema a tema (de los temas 01 al 09) dónde se fue aplicando en el código **y por qué se decidió implementarlo de esa forma**. Toda decisión tiene que estar argumentada con razones técnicas que giren alrededor de los pilares de la programación orientada a objetos y cuestiones de mantenibilidad, código limpio, flexibilidad al cambio, experiencia de usuario, accesibilidad, uso seguro, rendimiento y eficiencia.
 - ☐ Suma identificar pros y contras, si los tienen en mente.
 - ☐ El objetivo es que demuestren que saben lo que hacen y que tomaron decisiones con criterio y no mecanizadas.
 - ☐ Si se utilizó alguna biblioteca externa también se deberá justificar la elección.
- ☐ **Propuesta de valor agregado para promoción:** En esta sección se explicará y justificará la funcionalidad adicional propuesta para el punto de promoción.

Condiciones mínimas de aprobación

Para alcanzar la aprobación (**nota 4**) se deberán cumplir todas las siguientes pautas:

- ☐ Respetar **TODAS** las reglas de estilo de la cátedra y buenas prácticas indicadas en clase. Se corregirán todas las pantallas (visualización) y calidad de código según las exigencias de la cursada.
- ☐ Compilar sin errores ni advertencias (sí se admiten sugerencias del IDE).
- ☐ Debe resolver **TODAS** las necesidades del cliente (planteadas en la consigna y en el requerimiento) y **NUNCA** tener errores en el tiempo de ejecución.
- ☐ Separar de forma física (distintos proyectos) la capa de presentación (interfaz de usuario) de la lógica de negocio (entidades).
- ☐ Los datos de uso que no se cargan manualmente deben encontrarse pre-cargados (como literales) así como algunos datos históricos que simulan ejecuciones previas del programa, ya que no contaremos con persistencia de datos.
- ☐ Para todos los formularios se debe poder maximizar, minimizar y cambiar el tamaño de la ventana.
- ☐ Las clases y sus miembros deberán estar correctamente documentados con la herramienta de documentación XML o ser lo suficientemente autodescriptivos (esto último queda a criterio del docente corrector).
- ☐ Validar **todos** los ingresos de datos (cuando corresponda) mostrando mensajes claros para el usuario cuando un dato sea inválido.
- ☐ Abstractar las entidades y realizar un diseño orientado a objetos. Aplicar todos los temas de los temas 01 al 09 incluida.
- ☐ Utilizar al menos dos formularios.
- ☐ Se valorará el uso justificado de formularios **MDI**, pantallas modales y no-modales, menú de opciones.
- ☐ Ninguna entidad se debe comparar por defecto, si no por uno o varios de sus atributos, idealmente por su identificador. Se deberá cambiar el comportamiento de todos los métodos de comparación (método **Equals** y método **GetHashCode**).
- ☐ Todas las entidades sobrescribirán el método **ToString** y retornarán una cadena de texto con los datos que sean necesarios del objeto según el modelo utilizado.
- ☐ Al menos una jerarquía de herencia que aproveche el pilar del polimorfismo.
- ☐ Todas las entidades deberán estar correctamente encapsuladas exponiendo sólo sus operaciones y características esenciales, protegiendo el acceso y modificación libre de datos, y ocultando los detalles de la implementación.
- ☐ Al menos una sobrecarga de constructores y una sobrecarga de métodos.
- ☐ Clases, atributos y métodos que no correspondan o trabajen con el estado de una instancia particular deberán ser estáticos.
- ☐ Todos los objetos deberán inicializar su estado con los mínimos valores necesarios para que no exista lugar a fallos en el uso del objeto, no debiendo permitir que se instancien de otra forma.
- ☐ Declaración y uso de al menos dos enumerados.
- ☐ Declaración y uso de al menos dos tipos de colecciones genéricas distintas.
- ☐ Uso justificado de al menos una clase abstracta.

Si no se cumplen TODAS las condiciones mínimas de aprobación, no se continuará con la corrección y la nota será un dos (desaprobado).

Condiciones mínimas para promocionar

Para promocionar (**nota 6 o más**) se deberá cumplir todas las condiciones mínimas de aprobación, agregar las funcionalidades descritas más abajo y proponer una nueva funcionalidad en base al contexto del negocio.

La nueva funcionalidad deberá ser agregada a la aplicación siguiendo todos los criterios de calidad y buenas prácticas antes nombrados.

Justificar el valor agregado de su elección en la sección "*Propuesta de valor agregado*" del archivo README.md.

Además se deberá agregar:

- ☐ Tener la posibilidad de cancelar el cierre de la aplicación.
- ☐ Herencia gráfica (herencia de formularios)
- ☐ Para todos los formularios:
 - ☐ La posición y/o tamaño de los controles deberá ajustarse con la ventana.
 - ☐ Deberá existir un límite mínimo para ajustar el tamaño que será aquel donde se pierde visibilidad de los controles o se dificulta el trabajo con la aplicación.
- ☐ Todos los formularios deberán contar con un botón de ayuda que explique brevemente cómo se utilizan las funcionalidades.
- ☐ Agregar sobrecarga de operadores, indexadores y los implícitos y explícitos.

Si no se cumplen TODAS las condiciones mínimas de promoción, se corregirá el parcial pero la nota no podrá superar el cuatro (aprobado sin promoción).

IMPORTANTE:

Una vez cumplan con las condiciones y en caso que el docente crea necesario se llamará **en cualquier momento de la cursada** para defender el proyecto de forma **oral** para contar cómo lo resolvieron. Para esto deberán contar con cámara y micrófono, pudiendo conectarse desde un celular si fuera necesario. La exposición será grabada y se deberá presentar libreta universitaria o documento de identidad.

El resultado de la exposición oral **puede anular la promoción** si se detecta que no conocen en profundidad el proyecto o no entienden lo que hicieron.

Por el contrario, una muy buena exposición sumará un punto a la calificación. Habrá casos donde no se sume el punto pero se mantenga la promoción.

Calificación

Una vez que se hayan superado las condiciones de aprobación, se procederá a definir la nota por tema evaluando un uso correcto y bien justificado de cada uno de los temas vistos entre la clase 01 y la 09.

A tener en cuenta

Reglas de estilo, buenas prácticas y calidad de documentación.

Algoritmos y resolución de problemas. Reutilización de código.

Elección de tipos de datos. Trabajo con cadenas de texto, formatos de fechas, etc.

Uso correcto y justificado de miembros estáticos.

Abstracción.

Diseño de clases y objetos (diagrama de clases, atributos, métodos, constructores).

Uso correcto y justificado de sobrecargas de métodos, operadores y constructores.

Uso correcto y justificado de colecciones.

Creatividad en el diseño de formularios. Uso correcto de controles y eventos.

Buena experiencia de usuario.

Encapsulamiento y uso correcto y justificado de modificadores de acceso.

Uso correcto y justificado de propiedades y enumerados.

Uso correcto y justificado de jerarquías de herencia, clases abstractas y selladas.

Uso correcto y justificado de polimorfismo.

Solución robusta, con validaciones de datos.

Investigación, instalación y uso de una biblioteca externa (paquete Nuget) que resuelva o ayude a resolver alguna necesidad de la implementación.

Investigación y uso de un control o controles de Windows Forms novedoso.

Defensa oral.

Consultas sobre el parcial

Se podrán responder consultas sobre el parcial durante las clases (NO ocupará toda la clase).

De existir necesidad de hacer alguna consulta en otro momento de la semana, no se garantizará una respuesta veloz ya que la misma quedará sujeta a disponibilidad del docente y/o sus ayudantes.

Todas las preguntas deberán realizarse por **SLACK como único medio y sin excepción por el canal # *primer_parcial***.