

newshore_

Solutions
for the Airline
Industry

PRUEBA DE INGRESO

newshore_

REGLAS GENERALES

Se debe cumplir con el análisis diseño e implementación de la solución en lenguaje C# (Framework o Core) orientado a la web con ASP.NET MVC. Los temas que se evaluarán son:

- Arquitectura de diseño (MVC, N-Capas, Onion)
- Programación orientada a objetos (Clases, Herencia, Polimorfismo)
- Front-end (HTML, CSS)
- Consumo Web APIs (REST)
- Versionamiento código (Github, Bitbucket, Azure DevOps, GitLab)
- Base de datos SQL Server.

Se deben aplicar principios **SOLID** (como mínimo aplicar los principios de responsabilidad única).

Se valorará positivamente el uso de los siguientes puntos:

- Control de excepciones (Try-Catch)
- Logging en aplicación (Log4Net, NLog)
- Inyección de dependencias (Windsor, Ninject)
- Test unitarios y de integración (MSTest, NUnit)
- GIT Advanced
 - Distintas ramas (crear una rama alterna a la rama principal con al menos un cambio visible)
 - Pull request (dejar una creada y agregar el link al README.txt)
- Nomenclatura de código ([Referencia](#))

La entrega debe ser realizada como una solución alojada en el repositorio de código de su elección, debe ser de acceso público para facilitar la revisión de la misma, adjuntar un README.txt con toda la información pertinente para poder hacer funcionar la solución o todos aquellos temas que no logró implementar de manera correcta.

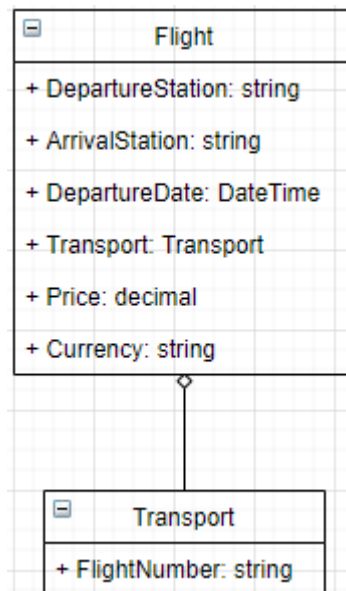
PROBLEMA 1 Modelado Clases

Descripción:

Como consultor encargado deberá diseñar y crear el componente de búsqueda de Vuelos del producto para la compañía que se venderá a las diferentes aerolíneas que son clientes de Newshore, por lo tanto, deberá codificar la aplicación según el modelo de clases dado y aplicar diseño a la plataforma.

A través de esta nueva plataforma, los futuros pasajeros podrán buscar los diferentes destinos a los que la aerolínea vuela, además de su horario y eventualmente el tiempo de vuelo.

Un usuario buscará la ciudad de partida y de llegada, seleccionando también una fecha para este viaje y según la disponibilidad de horario elegirá la hora en la que desea volar, cada vuelo según su horario tendrá un precio que será fijado por la aerolínea.



PROBLEMA 2 Maquetación web

Descripción:

Se necesita una interfaz web donde se vean representados los siguientes casos de uso:

Formulario de búsqueda de Vuelos Disponibles:

Interfaz con los campos de origen, fecha de vuelo y un botón CTA (Call to action) que realizará la acción de búsqueda de vuelos disponibles (según la respuesta de la Web API)

A Web Page

http://

Origen Destino

Buscar Vuelos

Resultado de Búsqueda de Vuelos:

Interfaz para visualizar los resultados de la búsqueda con base a la selección del usuario. Se debe tener en cuenta que la interface de búsqueda debe permanecer visible.

A Web Page

http://

Origen Destino

Buscar Vuelos

Disponibilidades de Vuelos

Origen	Destino	Fecha	Acciones
Bogota	Medellin	13/04/2018 09:00:00	Continuar
Bogota	Medellin	13/04/2018 09:30:00	Continuar
Bogota	Medellin	13/04/2018 09:40:00	Continuar
Bogota	Medellin	13/04/2018 11:00:00	Continuar

NOTA:

1. Para la solución es necesario hacer uso de CSS con una clara definición de las clases y elementos a utilizar.
2. Se valorará uso de frameworks orientados a la vista tales como Bootstrap, Materialize.
3. Los diseños presentados en los Mockups (Wireframes), corresponden a un diseño tentativo no es necesario la similitud a cabalidad de dicha realización de interfaz

Referencias:

<https://codepen.io/tranthanhrim1995/pen/ALgYLp>

PROBLEMA 3 Servicio API

Descripción:

De acuerdo a las clases modeladas en el Problema #1 y referente al paso #1 (Buscar viaje origen/destino, fecha salida). Las aerolíneas tienen rutas y fechas en las que operan sus vuelos, para este propósito se provee de una API que permita realizar la búsqueda de vuelos, que tiene todas las características e información necesaria para mostrar los vuelos en los que opera la aerolínea.

Para resolver el problema, se deberán tener en cuenta los siguientes datos: Origen, destino y fecha de salida. De acuerdo a estos datos que se solicitarán, consumir la API (<http://testapi.vivaair.com/otatest/api/values>) y recibir la información de vuelos en su respuesta

NOTA:

Formato de fecha debe ser yyyy-MM-dd (año-mes-día). Siempre una fecha de hoy hacia adelante.

Formato de rutas en código IATA, ejemplos:

MDE - Medellín

BOG - Bogotá

CTG - Cartagena

PEI - Pereira

Método para invocación de la API: *POST*

Entrada:

```
{
  "Origin": "BOG",
  "Destination": "CTG",
  "From": "2020-08-20"
}
```

Salida:

```
[
  {
    "DepartureDate": "2019-08-01T07:35:00",
    "DepartureStation": "MDE",
    "ArrivalStation": "BOG",
    "FlightNumber": "8020",
    "Price": 23094.0,
    "Currency": "COP"
  },
  {
    "DepartureDate": "2019-08-01T15:01:00",
    "DepartureStation": "MDE",
    "ArrivalStation": "BOG",
    "FlightNumber": "811",
    "Price": 23094.0,
    "Currency": "COP"
  }
]
```

PROBLEMA 4 Guardar resultado en BD**Descripción:**

Teniendo en cuenta todo el flujo anterior y luego de haber obtenido información desde la API, dichos resultados se pueden registrar en base de datos, según acción provocada por el usuario para este propósito. Por esto según el Mockup del Problema #2 se cuenta con un "enlace" o "botón" que se puede accionar con este fin.

Origen: Destino: Fecha:

Buscar Vuelos

Disponibilidades de Vuelos

Origen	Destino	Fecha	Acciones
Bogota	Medellin	13/04/2018 09:00:00	Continuar
Bogota	Medellin	13/04/2018 09:30:00	Continuar
Bogota	Medellin	13/04/2018 09:40:00	Continuar
Bogota	Medellin	13/04/2018 11:00:00	Continuar

PROBLEMA 5 Envío de Itinerario (opcional Perfil Senior – Edwin / Mauricio)**Descripción:**

Algoritmo para ver cómo resuelve

Propuesta 1: Modificar nombres de pasajeros / datos de contacto.

Propuesta 2: Descargar el itinerario (PDF)

Entrada:

Numero de reserva

NR12345

Volver

Buscar

Salida:

PNR	NR12345
Fecha de Ida	01/12/2018
Fecha de Vuelta	02/12/2018
Pasajero 1	Pepe Perez
Pasajero 2	Petra Perez
Numero de Contacto	311 000 00 00
Email de Contacto	prueba@test.com.co