

Prueba de ingreso FullStack

Andrés Felipe Hernández

<https://felipedelosh.github.io/AFE-PORTFOLIO/>

Newshore Prueba_Ingreso_Fullstack_2022

“La prueba descrita a continuación es por parte de un programador junior (Python + Django) pido disculpas debido a que esta es mi primera solución en (.net + Angular)”

Introducción:

A continuación se procede a presentar la solución al problema descrito en el PDF

La empresa NEWSHORE AIR necesita una solución web para poder conectar viajes a través del mundo, esta solución debe recibir como parámetros el origen y el destino de viaje del usuario, después el sistema debe consultar todos los vuelos asociados que tenga, y devolver la ruta de viaje al usuario si la ruta es posible o un mensaje avisando que la ruta no puede ser calculada.

1 -> Se crea un API en .net con 4 capas:

<https://github.com/felipedelosh/newShoreAIR>

- Bussines: es la encargada de la lógica del api, mapeo.
- Helper: es la encargada de consumir el api externa, almacenar/recuperar la cache, Autenticar los tokens, Calcular las rutas y calcular la conversión de monedas.
- Models: Es la encargada de conectar los modelos con la Base de datos? (En este caso la prueba no menciona dicha base de datos), Es la encargada de Almacenar los datos obtenidos del API.
- API: es la encargada de exponer las URLs al usuario.

2 -> Se crea un FRONTEND en ANGULAR:

<https://github.com/felipedelosh/newShoreAIRFRONTEND>

- Consume la API en .net
- Recibe/Manda Solicitudes al API .net

Nota: instalar el siguiente plugin para ejecutar el Frontend “error de CORPS” [06]

“Moesif Origin & CORS Changer”

- Descargar para firefox:

<https://addons.mozilla.org/es/firefox/addon/moesif-origin-cors-changer1/>

- Descargar para Chrome:

<https://chrome.google.com/webstore/detail/moesif-origin-cors-change/digfbfaphojjndkpccljibejjbppifbc>

Una vez descargado el Plugin SE TIENE QUE ACTIVAR sin él es imposible que el FRONTEND funcione.

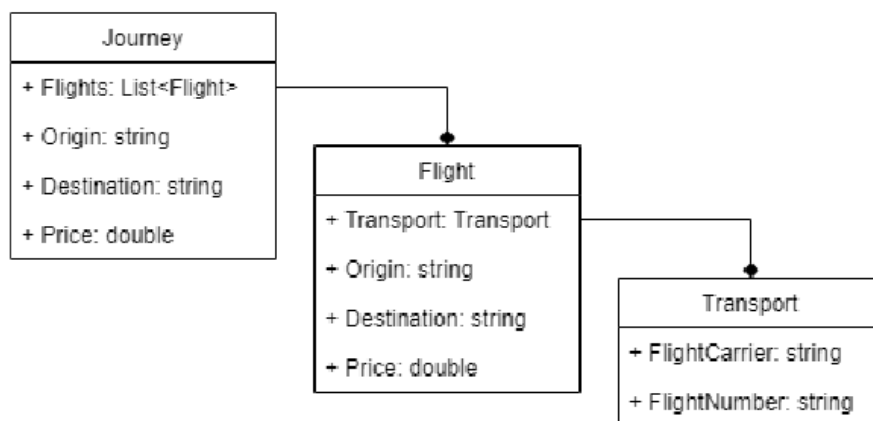
Objetivos de la Prueba:

Objetivo	Estado
Arquitectura de la solución	OK se adjunta en el documento.
Programación orientada a objetos	Ok se adjunta en el documento.
Consumo REST API	OK tanto del API como el FRONTEND
Control de versiones	OK se adjuntan LINKS GITHUB
Manejo de estilos en cascada CSS	OK se evidencian en el FRONTEND
Logs de aplicación	OK se hace el demo en el API (Nota: no se cubre toda debido a falta de tiempo.)
Control de excepciones	Ok se evidencias TRY CATCH en el API
Inyección de dependencias	OK se hace en el API con NUGGET AUTOFAC
Test unitarios	FAIL no conozco el uso de test.
Uso de states manager	FAIL no conozco el concepto o uso.
Uso de interceptors	FAIL no conozco el concepto o uso.
Uso de injection tokens	OK se verifica mediante llamadas al API bearer token simulado como en RubyOnRails.
Nomenclatura estándar	OK se evidencia en el API.
Cache	OK se evicencia en API.Helper.Cache

Análisis de los Enunciados:

Problema 1 – Modelado Clases

Como encargado de la solución deberá modelar un objeto estándar para trabajar con los vuelos en su aplicación, el modelo propuesto es el siguiente (diseño UML).



Se evidencia en la Capa Models del API

Problema 2 – Consumo REST API

Para poder calcular la ruta requerida por el usuario es necesario acceder a los vuelos asociados a NEWSHORE AIR, para este propósito se provee de una API que permita realizar la búsqueda de estos. Consuma la siguiente API (<https://recruiting-api.newshore.es/api/flights>) usando el método **GET** y use la clase **Flight** modelada en el **Problema 1** para mapear la respuesta.

El consumo se evidencia en API.Helper. GetAPIData.cs

Problema 3 – Obtener Ruta Parte 1

Se debe exponer un API en la cual el usuario pueda pasar sus parámetros de búsqueda. Exponga un API con un método **GET** que implemente el siguiente contrato.

Se expone el API en la siguiente dirección:

https://localhost:7036/api/v1/Journey/Get?origin=MZL&destination=CTG&Authorization=bearer kmzwa8awaa&Currience_selector=COP

Problema 4 – Obtener Ruta Parte 2

Se debe facilitar un formulario web donde el usuario pueda ingresar los campos **Origin** y **Destination**, dicho formulario debe cumplir las siguientes reglas.

- Los campos no pueden contener el mismo valor.
- Los campos solo deben permitir 3 caracteres exactamente.
- Los campos deben tener una estrategia para que el input siempre se registre en mayúscula.

Cuando el usuario ingrese los campos correctamente, debe existir un botón que se encargue de consumir el API expuesta en el **Problema 3** y mostrar el resultado en pantalla.

No funcional en su totalidad, solo se logró debido a una implementación (NO-BIDIRECCIONAL) del algoritmo BELLMAN FORD, solo es funcional para la siguiente ruta:

<https://recruiting-api.newshore.es/api/flights/0>

Problema 5 – Selección Moneda

Por último, después de obtener la respuesta en el **Problema 4** el usuario debe tener la posibilidad de ver el valor de la ruta en la moneda que seleccione, por defecto se asume que los valores devueltos en el **Problema 2** se encuentran en **USD**. Adicione al menos dos formatos más de moneda y la posibilidad de que al obtener la ruta de vuelo pueda cambiar esta y ver los valores equivalentes.

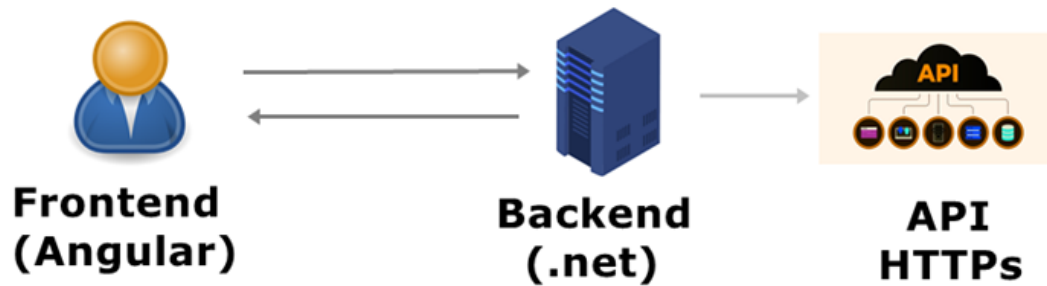
Nota: Recuerde que la API devuelve un valor genérico en **USD**, por lo tanto, la estrategia de conversión queda a decisión del desarrollador.

Se hace mediante API.helper. CurriencesConverter

Solo es un simulador de obtener la tasa de cambio (desde dólares USD) y multiplicar.

BACKEND

Arquitectura general:



Arquitectura General API:



Descripción general de las capas:

Capa de negocio (Business):

“Lleva toda la lógica de: Peticiones API, Mapeo Objetos API, Peticiones de Rutas”

- AvailabilityBusiness
- Mapper

Capa de Ayuda (helper):

“Realiza la Carga/Procesamiento de los Datos simula Gemas en RubyOnRails”

- Cache
- Routes Calculator.
- Token Authentication
- Curriences Converter

Capa de modelos (Models):

“Conector a persistencia de Datos”

- Contratos de Mapeo
- Atributos Objetos Json
- Objetos específicos.

Capa de API (newShoreAPI)

“Expone url al usuario”

- Inyección de dependencias.

IMPORTANTE: Instalar NUGGETS

Microsoft.AspNetCore.Cors

NLog.Web.AspNetCore

Microsoft.Extensions.Configuration

NewtonSoft.Json

AutoFac

AutoMapper

Para la capa Helper (Consumir API)

System.Text.Json = 7.0.0.0

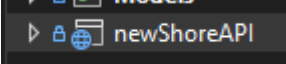
Esto evita el siguiente error:

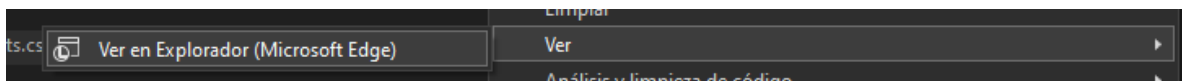
Response body

```
API_GET_ERR: System.Reflection.TargetInvocationException: Exception has been thrown by the target of an invocation.
--> System.IO.FileNotFoundException: Could not load file or assembly 'System.Text.Json, Version=7.0.0.2, Culture=neutral, PublicKeyToken=cc7b13ffcd2dd511'. El sistema no puede encontrar el archivo especificado.
File name: 'System.Text.Json, Version=7.0.0.2, Culture=neutral, PublicKeyToken=cc7b13ffcd2dd511'
at RestSharp.Serializers.Json.SystemTextJsonSerializer..ctor()
at System.RuntimeType.CreateInstanceDefaultCtor(Boolean ignoreCase)
--- End of inner exception stack trace ---
at System.RuntimeType.CreateInstanceDefaultCtor(Boolean ignoreCase)
at System.Activator.CreateInstance[T]()
at RestSharp.Serializers.SerializerConfig.<C_5`1>.UseSerializer(b_5_0())
at RestSharp.Serializers.SerializerConfig.UseSerializer(Func`1 serializerFactory)
at RestSharp.Serializers.SerializerConfig.UseSerializer[T]()
at RestSharp.Serializers.SerializerConfig.UseDefaultSerializers()
at RestSharp.RestClient.ConfigureSerializers(Configuration configuration, Configuration configuration)
at RestSharp.RestClient..ctor(RestClientOptions options, Configuration configuration, Configuration configuration, Configuration configuration, Boolean useClientFactory)
at RestSharp.RestClient..ctor(Uri baseUrl, Configuration configuration, Configuration configuration, Configuration configuration, Configuration configuration, Boolean useClientFactory)
at RestSharp.RestClient..ctor(String baseUrl, Configuration configuration, Configuration configuration, Configuration configuration, Configuration configuration, Configuration configuration)
at Helper.GetAPIData.GetHTTPServiceVrX(String url, String v) in C:\Users\felipe\source\repos\newShoreAIR\Helper\GetAPIData.cs:line 11
```

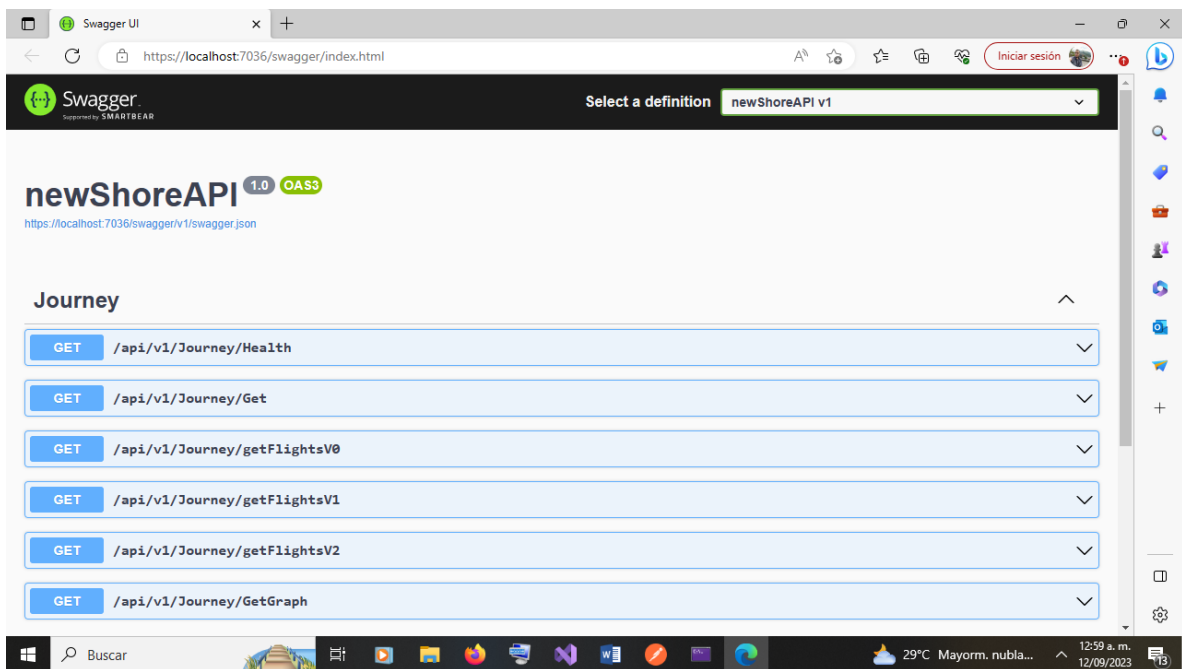
Download

Iniciar el servidor:

Clic derecho sobre: API  y luego ver e iniciar con EDGE.



Se nos abrirá la terminal y el navegador con la interface de SWAGGER



Se recomienda usar POSTMAN para hacer las pruebas.

https://localhost:7036/api/v1/Journey/Health

GET https://localhost:7036/api/v1/Journey/Health

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (4) Test Results

200 OK 386 ms 212 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "200",
3   "metadata": "v1.0",
4   "data": "the server is run :)"
5 }
```

<https://localhost:7036/api/v1/Journey/Health>

https://localhost:7036/api/v1/Journey/getFlightsV0

GET https://localhost:7036/api/v1/Journey/getFlightsV0

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
-----	-------	-------------	-----	-----------

Body Cookies Headers (4) Test Results 200 OK 3.30 s 1.04 KB Save as Example

Pretty Raw Preview Visualize JSON

```
1  "status": "200",
2  "metadata": "OK",
3  "data": [{"DepartureStation": "MZL", "ArrivalStation": "MDE", "FlightCarrier": "CO",
4           "FlightNumber": "8001", "Price": 200.0}, {"DepartureStation": "MZL",
           "ArrivalStation": "CTG", "FlightCarrier": "CO", "FlightNumber": "8002", "Price": 200.0},
           {"DepartureStation": "PEI", "ArrivalStation": "BOG", "FlightCarrier": "CO",
           "FlightNumber": "8003", "Price": 200.0}, {"DepartureStation": "MDE",
           "ArrivalStation": "BCN", "FlightCarrier": "CO", "FlightNumber": "8004", "Price": 500.0},
           {"DepartureStation": "CTG", "ArrivalStation": "CAN", "FlightCarrier": "CO",
           "FlightNumber": "8005", "Price": 300.0}, {"DepartureStation": "BOG",
           "ArrivalStation": "MAD", "FlightCarrier": "CO", "FlightNumber": "8006", "Price": 500.0},
           {"DepartureStation": "BOG", "ArrivalStation": "MEX", "FlightCarrier": "CO",
           "FlightNumber": "8007", "Price": 300.0}]
```

<https://localhost:7036/api/v1/Journey/getFlightsV0>

https://localhost:7036/api/v1/Journey/Get?origin=MZL&destination=CTG&Authorization=bearer...

GET https://localhost:7036/api/v1/Journey/Get?origin=MZL&destination=CTG&Authorization=bearer kmzw...

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> origin	MZL			
<input checked="" type="checkbox"/> destination	CTG			
<input checked="" type="checkbox"/> Authorization	bearer kmzwa8awaa			
<input checked="" type="checkbox"/> Currience_selector	COP			
Key	Value	Description		

Body Cookies Headers (4) Test Results 200 OK 142 ms 470 B Save as Example

Pretty Raw Preview Visualize JSON

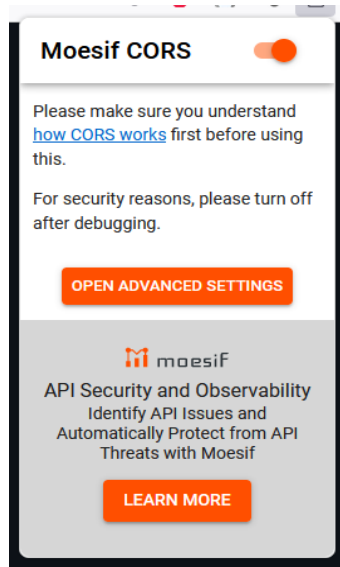
```
1  "status": "200",
2  "metadata": "v1.0",
3  "data": [{"Flights": [{"Transport": {"FlightCarrier": "CO", "FlightNumber": "8002"},
4           "Origin": "MZL", "Destination": "CTG", "Price": 200.0}, {"Message": "Find 2
           data:\n\nconversion price.\n\nReturn value in COP\n\n", "Origin": "MZL", "Destination": "CTG",
           "Price": 803366.0}]}
```

[https://localhost:7036/api/v1/Journey/Get?origin=MZL&destination=CTG&Authorization=bearer
kmzwa8awaa&Currience_selector=COP](https://localhost:7036/api/v1/Journey/Get?origin=MZL&destination=CTG&Authorization=bearer%20kmzwa8awaa&Currience_selector=COP)

FRONTEND

Descargar el proyecto desde: <https://github.com/felipedelosh/newShoreAIRFRONTEND>

Instalar el PLUGIN para el error CORPS y activar:



Iniciar el servidor: ng serve

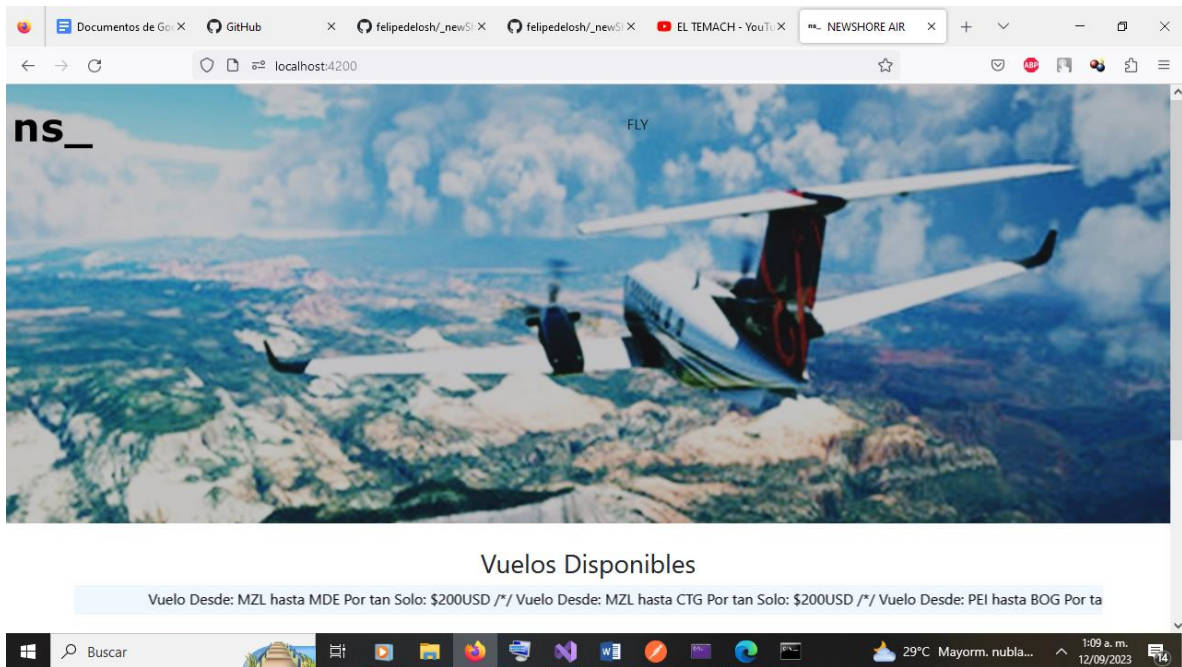
```
Seleccinar ng serve
C:\Users\felipedelosh\Documents\newShore\newShoreAIRFRONTEND>ng serve
✓ Browser application bundle generation complete.

Initial Chunk Files | Names          | Raw Size
vendor.js           | vendor         | 2.03 MB
styles.css, styles.js | styles        | 497.49 kB
scripts.js          | scripts        | 478.86 kB
polyfills.js        | polyfills      | 333.20 kB
main.js             | main           | 36.56 kB
runtime.js          | runtime        | 6.54 kB
                    | Initial Total  | 3.35 MB

Build at: 2023-09-12T06:08:32.198Z - Hash: 503f58fd683531b5 - Time: 15857ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

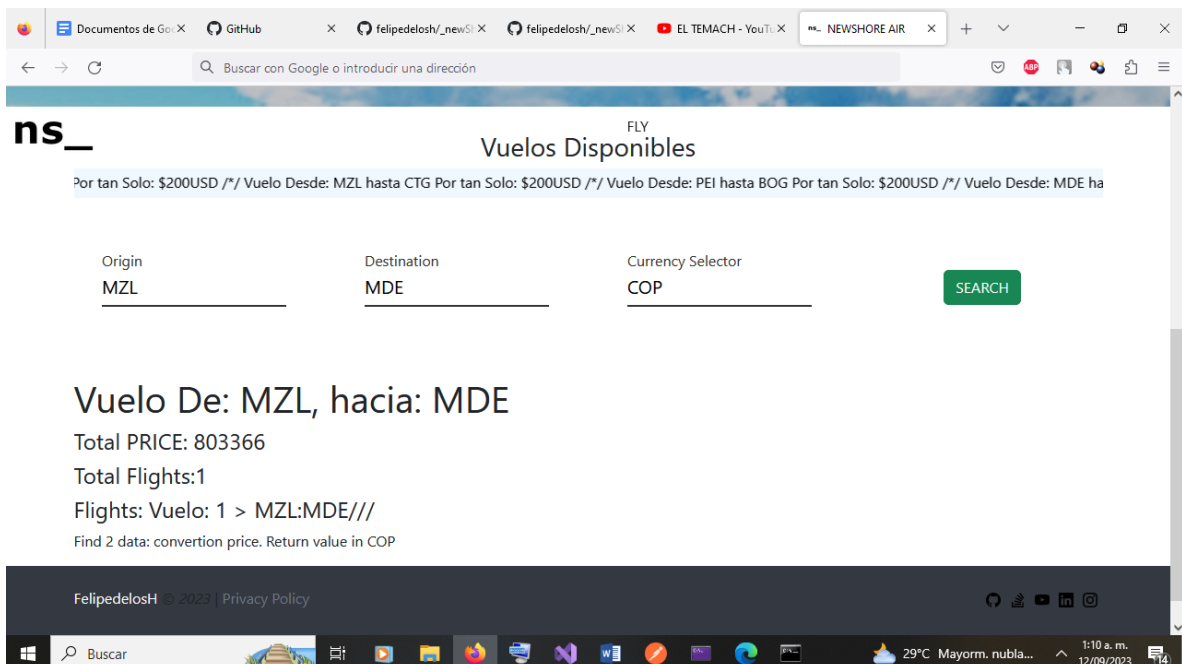
✓ Compiled successfully.
```

Ir al navegador a la dirección: <http://localhost:4200/>



Si todo sale bien podremos ver un texto con el nombre de los vuelos.

Vamos a realizar unas pruebas en el área de vuelo:



Manizales a Medellín en Pesos Colombianos.

ns_ FLY Vuelos Disponibles

tan Solo: \$300USD /*/ Vuelo Desde: BOG hasta MAD Por tan Solo: \$500USD /*/ Vuelo Desde: BOG hasta MEX Por tan Solo: \$300USD /*/

Origin: MZL Destination: BCN Currency Selector: MXN SEARCH

Vuelo De: MZL, hacia: BCN

Total PRICE: 12250

Total Flights: 2

Flights: Vuelo: 1 > MZL:MDE///Vuelo: 2 > MDE:BCN///

Find 3 data: convection price. Return value in MXN

FelipedelosH © 2023 | Privacy Policy

Manizales a Barcelona en Pesos mexicanos.

ns_ FLY Vuelos Disponibles

Por tan Solo: \$200USD /*/ Vuelo Desde: MDE hasta BCN Por tan Solo: \$500USD /*/ Vuelo Desde: CTG hasta CAN Por tan Solo: \$300USD /*/ Vuelo Desde: BOG

Origin: AWS Destination: KKK Currency Selector: PEN SEARCH

Vuelo De: AWS, hacia: KKK

Total PRICE: 0

Total Flights:

Flights:

FelipedelosH © 2023 | Privacy Policy

Valores incorrectos.

Referencias:

[01]Arquitectura en capas:

<https://learn.microsoft.com/es-es/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>

[02]Crear solución en blanco desde Github

<https://learn.microsoft.com/es-es/visualstudio/get-started/tutorial-open-project-from-repo?view=vs-2022>

[04]onvertir Json en Object con Newton Json

<https://www.techieclues.com/blogs/convertng-json-to-string-in-csharp>

[05]Consumir Json desde otras APPs

<https://www.c-sharpcorner.com/article/enable-cors-consume-web-api-by-mvc-in-net-core-4>

[06] Problema CORPS

<https://www.enmilocalfunciona.io/entendiendo-cors-y-aplicando-soluciones/>