

Fundamentos de Informática

Caso de Negocio

La aerolínea **DreamFly** es una aerolínea local que realiza sólo vuelos domésticos dentro de su país.

La empresa es pequeña comparada con las multinacionales, y cuenta con solo 6 aeronaves. Pero su CEO quiere llevar a su aerolínea a la categoría premium y brindar un **servicio “world class”**.



El mercado de las aerolíneas, en los últimos años se viene desatando una competencia feroz para ganar **cuota de mercado**, y en este sentido, la tecnología es una aliada estratégica para que las empresas estén cada vez más cerca de sus clientes con el objetivo de brindarles mejores productos y servicios.

En este contexto, el CTO de **DreamFly** solicita de nuestros servicios para **desarrollar un sistema** que permita gestionar las **dietas y restricciones** que maneja la aerolínea.

Los clientes, para poder **elegir los menú con dietas especiales**, **deben abonar** una **tarifa adicional** a su pasaje.



En ese sentido, el CTO comenta que, por el momento, los tipos de **dieta y restricciones** con las que opera la aerolínea son las siguientes:

DIETA

Código	Restriction (Eng)	Restriccion (Es)	Stock	Expiration	USD
R001	Gluten-free	Libre de Gluten	350	12/08/2024	100
R002	Dairy-free	Libre de Lácteos	240	24/07/2024	112
R003	Nut-free	Libre de Frutos Secos	160	22/12/2024	123
R004	Vegetarian	Vegetariano	430	14/07/2024	144
R005	Vegan	Vegano	224	15/07/2024	155

Para los pasajeros internacionales, se ofrece la **descripción en idiomas Español e Inglés**, además de la **tarifa en Dólares**.



Se solicita,

Desarrollar una API para gestionar las DIETAS con las siguientes especificaciones:

- 1) El base_URL de la API será: “<host>:<port>/api/dreamfly”
- 2) Se debe almacenar el listado de “**DIETA**” de una forma **persistente**.
- 3) Crear un endpoint tipo **GET** con la siguiente URi “/dietas”, que devuelva **todas las dietas** de la compañía
- 4) Crear un endpoint tipo **POST** con la siguiente URi “/dietas”, que permita **agregar** una nueva dieta, por ejemplo:

```
{  
  "id":6,  
  "code":"R006",  
  "date":"21/09/2024",  
  "name_eng":"Egg-free",  
  "name_esp":"Libre de Huevos",  
  "stock":220,  
  "usd_value":140  
}
```

- 5) Para los clientes de **Argentina**, que quiera consultar a **una única dieta**, habilitar un endpoint **GET** que traiga **solo el texto en castellano** y la **cotización en pesos**:

“/dietas/<id>/pais/ARG”

En caso que el **parámetro <id>** no venga definido, devolver un **código de error HTTP 400** indicando que el parámetro es obligatorio.

Hints:

- Utilizar Flask para desarrollar la API
- Utilizar Postman como cliente de su API
- Estructurar el proyecto correctamente y se valora el uso de comentarios
- Usar la Base de Datos como medio de persistencia
- Utilizar POO
 - definir en él/los objetos el método
 - “serialize(self)”
 - “constructor” y “destructor”

Para la entrega:

- Todos los archivos deberán ser comprimidos utilizando zip, rar u otra.
 - “nombre_apellido_fdi_final_04_07_2024.zip”