

Prueba Backend

- **Entorno de Desarrollo**

Para el desarrollo de la aplicación se decidió trabajar con Python en contenedores de Docker, Uno el cual contiene la lógica de la aplicación realizada con el microframework Flask y otro en el cual se encuentra la base de datos.

Para iniciar el entorno de desarrollo, la importación de la base de datos y ejecución de pruebas, ver el README del repositorio.

Nota: Para la importación de la base de datos, si no funciona es necesario hacerlo a través de **adminer**, esto ya que probe en diferentes computadores y no me funciono la importación de la misma forma

- **Código**

Desde el comienzo del desarrollo de la API se buscó trabajar con el principio de SOLID, DRY y el Zen de Python. Además de estar realizando las pruebas unitarias que probaran desde un inicio el funcionamiento del código. Sin embargo y debido a que el tiempo no me lo permitió, hay partes en el código que aún pueden ser mejoradas, de igual forma, hicieron falta algunas pruebas unitarias.

Cada endpoint (orders, products) tiene sus propios Blueprints y Controladores, separando de esta forma el proceso de cada uno y teniendo su propia responsabilidad.

Blueprints / API

Son los encargados de recibir y direccionar las peticiones, además de dar respuesta de las mismas.

Controladores

En los diferentes controladores se recibe la petición desde los blueprints y se procesan los datos recibidos de los modelos, información que luego de ser procesada es enviada de vuelta al usuario.

Helpers

Aquí algunas de las funcionalidades que se utilizan de manera general en los diferentes controladores y blueprints para dar respuesta de las peticiones.

Modelos

Aquí es donde se maneja la lógica del negocio. Comunicación directa a la base de datos que retorna información que se procesa en los controladores.

- Endpoints

URL: <http://192.168.55.199:5000>

1. Consultar qué productos y qué cantidad puede ser alistada desde el inventario.

GET - */products/organize/inventory*

Se realizó teniendo en cuenta la prioridad de los pedidos y si van o no para la misma persona

```
▼ "data": [  
  ▼ {  
    "id": 1,  
    "priority": 1,  
    "address": "KR 14 # 87 - 20",  
    "user": "Sofia",  
    "deliveryDate": "2019-03-01",  
    ▼ "products": [  
      ▼ {  
        "id": 1,  
        "name": "Leche",  
        "quantity": 1,  
        "provider_id": 1  
      },  
      ▼ {  
        "id": 2,  
        "name": "Huevos",  
        "quantity": 3,  
        "provider_id": 1  
      },  
      ▼ {  
        "id": 3,  
        "name": "Manzana Verde",  
        "quantity": 7,  
        "provider_id": 1  
      },  
      ▼ {  
        "id": 4,  
        "name": "Pepino Cohombro",  
        "quantity": 5,  
        "provider_id": 1  
      },  
      ▼ {  
        "id": 28,  
        "name": "Tomate Larga Vida Maduro",  
        "quantity": 1,  
        "provider_id": 2  
      }  
    ]  
  }  
]
```

2. Consultar los productos que deben ser alistados por transportadores, y a qué transportador le corresponde cada pedido.

GET - /products/organize/providers

Se realizó teniendo en cuenta la prioridad de los pedidos y si van o no para la misma persona

```
{
  "status": "success",
  "data": [
    {
      "id": 1,
      "priority": 1,
      "address": "KR 14 # 87 - 20",
      "user": "Sofia",
      "deliveryDate": "2019-03-01",
      "products": [
        {
          "id": 2,
          "name": "Huevos",
          "quantity": 18,
          "provider_id": 1,
          "provider": {
            "id": 1,
            "name": "Ruby"
          }
        },
        {
          "id": 3,
          "name": "Manzana Verde",
          "quantity": 3,
          "provider_id": 1,
          "provider": {
            "id": 1,
            "name": "Ruby"
          }
        },
        {
          "id": 37,
          "name": "Pan Bimbo",
          "quantity": 7,
          "provider_id": null
        }
      ]
    }
  ]
}
```

3. Productos más / menos vendidos el día 1 de marzo

GET - /products/sold

Parameters:

- **sort:** (string) asc o desc
Ordenar productos por ventas. Default **asc**

- **limit:** (number)
Traer mayor cantidad de productos. Default 3

```
{
  "status": "success",
  "data": [
    {
      "product_id": 27,
      "name": "Piment\\u00f3n Verde",
      "quantity": 1
    },
    {
      "product_id": 41,
      "name": "banano",
      "quantity": 1
    },
    {
      "product_id": 29,
      "name": "Cilantro x 500grs",
      "quantity": 1
    }
  ],
  "error": ""
}
```

4. Dado el Id de un pedido, saber qué productos y qué cantidad pueden ser alistados según sistema de inventario y cuáles deben ser abastecidos por los proveedores.

GET - /orders/{order_id}/stock

```
{
  "status": "success",
  "data": {
    "in_stock": [
      {
        "product_id": 16,
        "name": "Acelga",
        "quantity": 9
      }
    ],
    "out_of_stock": [
      {
        "product_id": 16,
        "name": "Acelga",
        "quantity": 1491
      }
    ]
  },
  "error": ""
}
```

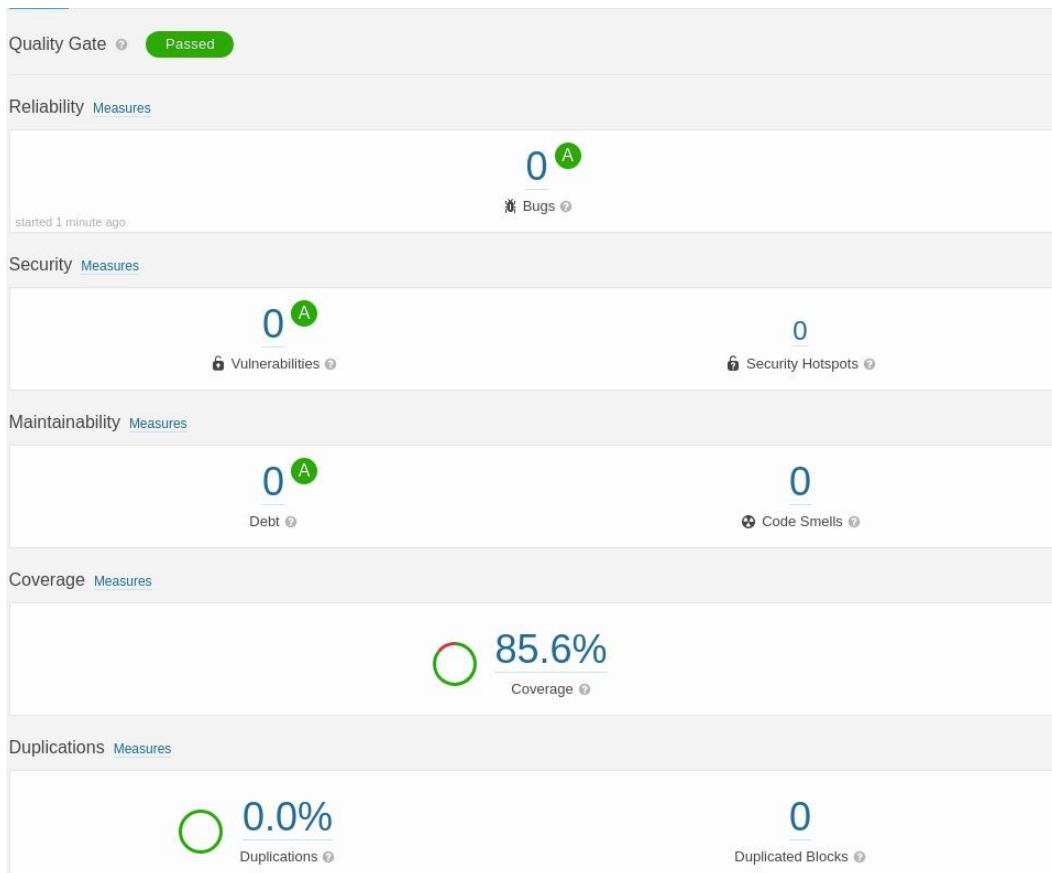
5. Calcular el inventario del día 2 de marzo, teniendo en cuenta las pedidos despachados el 1 de marzo.

GET - /products/inventory

```
{
  "status": "success",
  "data": [
    {
      "id": 1,
      "name": "Leche",
      "quantity": 2
    },
    {
      "id": 2,
      "name": "Huevos",
      "quantity": 0
    },
    {
      "id": 3,
      "name": "Manzana Verde",
      "quantity": 0
    },
    {
      "id": 4,
      "name": "Pepino Cohombro",
      "quantity": 1
    },
    {
      "id": 5,
      "name": "Piment\u00f3n Rojo",
      "quantity": 0
    }
  ]
}
```

- **Pruebas**

En el desarrollo de la API se realizaron las respectivas pruebas unitarias y se utilizó **Sonarqube** para las pruebas de código estático y validación del código.



- **Mejoras**

Debido a la falta de tiempo no pude implementar algunas mejoras que considero se requiere en la API.

Autenticación: Se puede realizar una implementación de usuarios sencilla por medio de Token o usando JWT.

Código: A pesar de que desde el comienzo trate de utilizar las mejores prácticas y trabajando con principios SOLID, para cierta parte del código aún se pueden realizar algunas mejoras.

Active Record: Para el correcto manejo de la base de datos, sería un punto positivo trabajar de esta forma la base de datos, sin embargo, no conozco ninguno que pueda implementar y por falta de tiempo no investigue.