**Aplicaciones restful**: Catalogo

**Objetivo de la aplicación**

El objetivo de la aplicación es que, si la consume un usuario final, este pueda ver los productos que están cargados, así como también subscribirse para recibir notificaciones cada ve que hay alguna actualización.

Por otro lado, la aplicación permite a quien la administre poder agregar, modificar y eliminar productos. También cuenta con la opción de poder enviar un archivo en formato Json para actualizar de forma masiva todos los productos que tenga cargados.

**Casos de uso**

1. Que un cliente pueda subscribirse para recibir notificaciones.
2. Que un cliente pueda desuscribirse para no recibir notificaciones.
3. Que un administrador pueda enviar notificaciones.
4. Que un administrador pueda crear, editar y eliminar productos.
5. Que un administrador pueda actualizar productos desde un archivo.

**Recursos del sistema**

1. Envió de Emails.
2. Subida de imágenes a servidor cloud Cloudinary.
3. Suida y bajada de archivos al servidor local.
4. Suscripción, desuscripcion y notificaciones.
5. Alta, baja y modificación de Productos.
6. Actualización de productos a través de archivos.

**Rutas del sistema**

[POST] /api/subscribe/{ email: 'prueba@prueba.com' }

Descripción: Permite subscribir un email si todavía no fue agregado.

* Nombre del parametro: Email
* Tipo del parámetro: String

**Success Response** {200 rta: 'Realizado }

**Error Response** {404 msg: el email ya esta agregado,

type: ERR\_EMAIL\_FOUND }

1. **[POST]** /api/unsubscribe/{ email: 'prueba@prueba.com' }

Descripción: Permite desubscribir un email si fue agregado.

* Nombre del parametron: Email
* Tipo del parámetro: String

**Success Response** { 200 rta: 'Eliminado'}

**Error Response** { 404 msg: el email no esta agregado.,

type: ERR\_EMAIL\_NOT\_FOUN }

1. **[POST]** /api/notify/{ subject: 'test', htmlMSG: 'test msg' }

Descripción: Permite notificar a todos los usuarios que estén subscriptos.

* Nombre del parametro: Subject
* Tipo del parametro: string
* Nombre del parametro: htmlMSG
* Tipo del parametron: string

**Success Response** { 200 rta: 'Realizado'}

1. **[GET]** /api/actualizar

Descripción: Devuelve un archivo .json en la carpeta 'output' con toda la colección de productos que se halle en la base de Mongo.

**Success response:** {200: Toda la colección exportada en un archivo correctamente}

**Error response**: {500: Error de servidor. Reintente operación}

1. **[GET]** api/actualizar/[nombre producto]

Descripción: Devuelve un archivo .json en la carpeta 'output' con la información de ese producto específico.

* Nombre del parametro: Name
* Tipo del parametro: string

1. **[PUT]** /api/actualizar/update/[nombre-de-archivo]

Descripción: El archivo con las actualizaciones que se quieren realizar de los productos debe estar en la carpeta input en formato .json. La API parseará el archivo y modificará los registros en la base de datos de Productos en MongoDB.

**Success response:** {200: Archivo actualizado correctamente}

1. [GET] /api/productos/ Codigo: String o nada consultatodo

Descripción: Consultar todos los productos de la base, o uno específico por código.

**Success response:** 200 y lista de productos

**Error response**: 500 y descripción de error

1. [Post] /api/productos/ Producto:objeto, Imagen:file

Descripción: Agregar un producto a la base

**Success response**: 200 y objeto producto dado de alta

**Error response**: 400 y descripción de error

500 y descripción de error

1. Get /api/productos /:id

Descripción: Buscar un producto por ID.

* Nombre del parametro: ID
* Tipo del parametro: string

**Success response**: 200 y objeto producto dado de alta

**Error response** 500 y descripción de error

1. Put /api/productos /:id

Descripción: Actualizar un producto.

* Nombre del parametro: ID
* Tipo del parametro: string
* Nombre del parametro: Producto
* Tipo del parametro: objeto
* Nombre del parametro: Imagen
* Tipo del parametro: file

**Success response**: 200 y mensaje de Ok

**Error response** 400 y descripción de error

500 y descripción de error

1. Delete /api/productos /:id ID:string

Descripción: Borrar un producto de la base

* Nombre del parametro: ID
* Tipo del parametro: string

**Success response**: 200 y mensaje de Ok

**Error response** 500 y descripción de error

**Códigos de Error**

Código Descripción

200 ok - Todo trabajo como fue esperado.

400 Data error – Datos incorrectos.

404 Not Found: - El ítem solicitado no existe.

500 Server Error: Algo salió mal.