**Análisis de requerimientos:**

*Requerimientos:*

*1.) Crear una API para una UI con los siguientes endpoints:*

*a. Obtener la información guardada del usuario (imagen, nombre, descripción).*

*▪ Leer y escribir la imagen del usuario.*

*▪ Leer y escribir la información del usuario en la base de datos.*

*b. Obtener los últimos 10 tweets del usuario.*

*c. Modificar la información del usuario de forma parcial (PATCH) (opcional).*

*d. Login (opcional)*

*e. Insertar un nuevo usuario (opcional)*

*f. Generar la documentación y ejemplos para el uso de la API -swagger/Postman- (opcional)*

*2.) Debe hacer uso de la API de twitter https://developer.twitter.com/en/docs/twitter-api para*

*obtener los Tweets.*

*3.) Los datos deben persistir, es decir, la API debe contar con una conexión a una base de datos.*

**ENTORNO DE DESARROLLO LOCAL:**

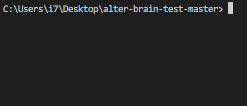
* + PHP v8.1.10
  + Composer v2.5.1
  + Laravel Framework v10.2.0
  + MySql v8.0.30
  + Visual Studio Code v1.74.3
  + Git v2.39.0.windows.2

**PASOS DE INSTALACION Y PRUEBAS:**

* Tenga en cuenta que debe tener estos requerimientos en su equipo local para poder instalar dependencias necesarias, ejecutar y probar la aplicación:
  + PHP v8+
  + Node JS v16+
  + Composer v2+
  + Gestor de base de datos MySql v8+

**INSTALACIÓN:**

1. Descargue y descomprima el ZIP si lo tiene disponible, en caso de no tenerlo descargue el repositorio en <https://gitlab.com/b-jrm/alter-brain-test/-/tree/master>
2. En la raíz de la carpeta ubique el archivo **.env.example** y cámbiele el nombre por **.env**
3. Cree la base de datos en su gestor local con el nombre **alter-brain-test** con la collation **utf8mb4\_unicode\_ci**
4. Ingrese en la ubicación de la carpeta desde la terminal CMD, y ejecute los siguientes comandos:

*Ubicación, En mi caso tengo la carpeta en el escritorio.*

***Comandos:***

*composer update*

*npm install*

*php artisan key:generate*

*php artisan storage:link*

*php artisan migrate --seed*

**¿COMO USAR LA API?**

**Endpoints Map:**

* **Guest:**
  + **(post)** <http://127.0.0.1:8000/login>
  + **(post)** <http://127.0.0.1:8000/register>
* **Auth:**
  + **(get)** <http://127.0.0.1:8000/api/users>
  + **(get)** [http://127.0.0.1:8000/api/users/{id}](http://127.0.0.1:8000/api/users/%7bid%7d)
  + **(post)** <http://127.0.0.1:8000/api/user/create>
  + **(post)** [http://127.0.0.1:8000/api/user/{id}/update](http://127.0.0.1:8000/api/user/%7bid%7d/update)
  + **(patch)** [http://127.0.0.1:8000/api/avatar/{id}/update](http://127.0.0.1:8000/api/avatar/%7bid%7d/update)
  + **(get)** [http://127.0.0.1:8000/api/tweets/{user\_id}](http://127.0.0.1:8000/api/tweets/%7buser_id%7d)
  + **(post)** <http://127.0.0.1:8000/logout>

|  |  |  |  |
| --- | --- | --- | --- |
| ENDPOINT Generar una sesión, creando y retornando el token, (Si ya tiene un token no es necesario) | | | <http://127.0.0.1:8000/api/login> |
| **METHOD** | **REQUEST** | **RESPONSE** | |
| POST | **Headers**:  **Accept:** application/json  **Params:**   **email:** (string)required  **password:** (string)required  **Ejemplo:**   **email:** ditech@test.com  **password:** Ditech2023 | {      "response": "Welcome!, Ditech",      "access": {          "type": "Bearer",          "token": "11|fFWVEAIrotguajzRHOPOxZmHGLZLVJUlLwz5JoTf"      }  } | |

|  |  |  |  |
| --- | --- | --- | --- |
| ENDPOINT Registrarme como un usuario,  Se crea y retorna el token de acceso | | <http://127.0.0.1:8000/api/register> | |
| **METHOD** | **REQUEST** | | **RESPONSE** |
| POST | **Headers**:  **Accept:** application/json  **Params:**   **name:** (string)required unique  **email:** (string)required unique  **password:** (string)required  **password\_confirmation:** (string)required | | {      "type": "success",      "message": "Registrado con éxito",      "response": {          "user": {              "name": "Mavega",              "email": "mauricio.vega@test.com",              "updated\_at": "2023-03-06T18:28:04.000000Z",              "created\_at": "2023-03-06T18:28:04.000000Z",              "id": 15          },          "access": {              "type": "Bearer",              "token": "12|cUSQILeWdU5vO0d9JxCVY3wqdxK7MO87xeBno6wg"          }      }  } |

|  |  |  |
| --- | --- | --- |
| ENDPOINT | METHOD HTTP | REQUEST |
| <http://127.0.0.1:8000/api/users>  Obtener todos los usuarios existentes de la base de datos | GET | **Autorization:**  **Type:** Bearer Token  **Token:** *personal\_access\_token*  **Headers**:  **Accept:** application/json |
| **RESPONSE** | [      {          "id": 1,          "nombre": "User Name 1",          "imagen": "http://127.0.0.1:8000/storage/avatars/default.png",          "descripcion": "User Default System"      },  {…},{…}  ] | |

|  |  |  |
| --- | --- | --- |
| ENDPOINT | METHOD HTTP | REQUEST |
| [http://127.0.0.1:8000/api/user/{id}](http://127.0.0.1:8000/api/user/%7bid%7d)  Obtener un usuario especifico de la base de datos por su ID | GET | **Autorization:**  **Type:** Bearer Token  **Token:** *personal\_access\_token*  **Headers**:  **Accept:** application/json |
| **RESPONSE  Ejemplo de endpoint:** <http://127.0.0.1:8000/api/user/8> | {      "type": "success",      "message": "Usuario encontrado",      "response": [          {              "id": 8,              "name": "bjrm",              "email": "bjrm@gmail.com",              "avatar": "http://127.0.0.1:8000/storage/avatars/default.png",              "description": "Usuario de pruebas",              "email\_verified\_at": **null**,              "created\_at": "2023-03-03T00:32:33.000000Z",              "updated\_at": "2023-03-06T16:36:51.000000Z"          }      ]  } | |

|  |  |  |  |
| --- | --- | --- | --- |
| ENDPOINT Crear un usuario en la base de datos | | <http://127.0.0.1:8000/api/user/create> | |
| **METHOD** | **REQUEST** | | **RESPONSE** |
| POST | **Autorization:**  **Type:** Bearer Token  **Token:** *personal\_access\_token*  **Headers**:  **Accept:** application/json  **Params:**  **name:** (string)required  **email:** (string)required unique  **description:** (string)required | | {      "type": "success",      "message": "Usuario creado",      "response": []  } |

|  |  |  |  |
| --- | --- | --- | --- |
| ENDPOINT Actualizar la información de un usuario especifico en la base de datos | | [http://127.0.0.1:8000/api/user/{id}/update](http://127.0.0.1:8000/api/user/%7bid%7d/update) | |
| **METHOD** | **REQUEST** | | **RESPONSE** |
| POST | **Autorization:**  **Type:** Bearer Token  **Token:** *personal\_access\_token*  **Headers**:  **Accept:** application/json  **Params:**  **name:** (string)required  **email:** (string)required unique  **description:** (string)required | | {      "type": "success",      "message": "Usuario (bjrm) actualizado correctamente",      "response": []  } |

|  |  |  |  |
| --- | --- | --- | --- |
| ENDPOINT Actualizar la imagen de un usuario por su ID | | [http://127.0.0.1:8000/api/user/{id}/update](http://127.0.0.1:8000/api/user/%7bid%7d/update) | |
| **METHOD** | **REQUEST** | | **RESPONSE** |
| PATCH | **Autorization:**  **Type:** Bearer Token  **Token:** *personal\_access\_token*  **Headers**:  **Accept:** application/json  **Body:**  **avatar:** file  **Ejemplo:**  <http://127.0.0.1:8000/api/avatar/8/update> | | {      "type": "success",      "message": "Imagen de usuario actualizada",      "response": [          {              "id": 8,              "name": "bjrm",              "email": "bjrm@gmail.com",              "avatar": "http://127.0.0.1:8000/storage/avatars/default.png",              "description": "Usuario de pruebas",              "email\_verified\_at": **null**,              "created\_at": "2023-03-03T00:32:33.000000Z",              "updated\_at": "2023-03-06T16:36:51.000000Z"          }      ]  } |

|  |  |  |  |
| --- | --- | --- | --- |
| ENDPOINT Obtener los tweets de un usuario por su ID | | [http://127.0.0.1:8000/api/tweets/{user\_id}](http://127.0.0.1:8000/api/tweets/%7buser_id%7d) | |
| **METHOD** | **REQUEST** | | **RESPONSE** |
| GET | **Autorization:**  **Type:** Bearer Token  **Token:** *personal\_access\_token*  **Headers**:  **Accept:** application/json  **Ejemplo:** <http://127.0.0.1:8000/api/tweets/8> | | {      "title": "Unauthorized",      "type": "about:blank",      "status": 401,      "detail": "Unauthorized"  }  (Falta Token de twitter – No disponible) (Si tiene disponible un token agregarlo en el archivo app/Http/Traits/Twitter => Linea 16) |

|  |  |  |  |
| --- | --- | --- | --- |
| ENDPOINT Expirar los tokens del usuario actual | | <http://127.0.0.1:8000/api/logout> | |
| **METHOD** | **REQUEST** | | **RESPONSE** |
| POST | **Autorization:**  **Type:** Bearer Token  **Token:** *personal\_access\_token*  **Headers**:  **Accept:** application/json | | {      "type": "success",      "message": "Ditech!, Session Finally",      "response": []  } |