**Documentación de Clases de Autenticación en CashFlowly**

**Controlador de Usuarios**

**RegistrarUsuario**

**Endpoint:** POST /api/usuarios/registrar

**Descripción:** Permite registrar un nuevo usuario en la aplicación y devuelve un token de autenticación.

**Parámetros:**

* usuarioDto (**UsuarioRegistroDto**): Objeto con la información del usuario a registrar.

**Respuesta:**

* 200 OK: Devuelve un token de autenticación si el registro es exitoso.
* 400 BadRequest: Si ocurre un error en la validación o el correo ya está en uso.

**Ejemplo de Respuesta Exitosa:**

{

"Token": "<jwt-token>"

}

**Login**

**Endpoint:** POST /api/usuarios/login

**Descripción:** Este método permite a los usuarios autenticarse en la aplicación mediante un LoginDto.

**Parámetros:**

* loginDto (**LoginDto**): Objeto que contiene el email y la contraseña del usuario.

**Respuesta:**

* 200 OK: Devuelve un token de autenticación si las credenciales son correctas.
* 400 BadRequest: Si el modelo enviado no es válido.
* 401 Unauthorized: Si las credenciales son incorrectas.
* 500 Internal Server Error: Si ocurre un error inesperado en el servidor.

**Ejemplo de Respuesta Exitosa:**

{

"Token": "<jwt-token>"

}

**DTOs (Data Transfer Objects)**

**LoginDto**

**Descripción:** Representa los datos necesarios para que un usuario pueda autenticarse en la aplicación.

**Propiedades:**

* Email (**string**): Dirección de correo del usuario.
  + **Restricciones:** Obligatorio y debe tener formato de correo electrónico válido.
* Password (**string**): Contraseña del usuario.
  + **Restricciones:** Obligatoria.

**Servicio de Autenticación**

**AuthService**

**Descripción:** Servicio que gestiona la autenticación y registro de usuarios en la plataforma CashFlowly.

**Dependencias:**

* UsuarioRepository: Repositorio para gestionar los usuarios.
* IConfiguration: Para obtener configuraciones de la aplicación.
* CashFlowlyDbContext: Contexto de base de datos para la persistencia.

**RegistrarUsuarioAsync**

**Descripción:** Registra un nuevo usuario en la base de datos y genera un token de autenticación si el registro es exitoso.

**Parámetros:**

* usuarioDto (**UsuarioRegistroDto**): DTO con los datos del usuario a registrar.

**Retorno:**

* string: Token JWT de autenticación generado para el usuario registrado.

**Errores:**

* Lanza una excepción si el correo ya está en uso.

**LoginAsync**

**Descripción:** Autentica a un usuario con su email y contraseña, y devuelve un token de autenticación.

**Parámetros:**

* loginDto (**LoginDto**): DTO con los datos de inicio de sesión.

**Retorno:**

* string: Token JWT si la autenticación es exitosa.

**Errores:**

* UnauthorizedAccessException si las credenciales son incorrectas.

**GenerarToken**

**Descripción:** Genera un token JWT para un usuario autenticado.

**Parámetros:**

* usuario (**Usuario**): Usuario autenticado.

**Retorno:**

* string: Token JWT generado.

**Claims incluidos en el Token:**

* sub: Email del usuario.
* jti: Identificador único del token.
* id: ID del usuario.
* nombre: Nombre del usuario.

**Interfaz del Servicio de Autenticación**

**IAuthService**

**Descripción:** Define la interfaz del servicio de autenticación en CashFlowly.

**Métodos:**

* Task<string> RegistrarUsuarioAsync(UsuarioRegistroDto usuarioDto): Registra un usuario y devuelve un token JWT.
* Task<string> LoginAsync(LoginDto loginDto): Autentica a un usuario y devuelve un token JWT.