# Micro servicio de autenticación con .NET Core Web api siguiendo los principios de Clean Arquitecture y DDD.

## Visión global

El proyecto se basa en desarrollar un micro servicio de registro de usuario, confirmación de email válido, posterior inicio de sesión y recuperación de contraseña si la ha olvidado utilizando .NET Core Web API, JSON Web Token (JWT), un servicio de correo electrónico. Todo ello siguiendo el enfoque de arquitectura limpia y diseño guiado por el dominio.

## Pasos a seguir

1. Definir los casos de uso: Se deben definir los casos de uso que cumplirá el micro servicio, como el registro de usuario, la confirmación de email válido, el inicio de sesión y la recuperación de contraseña.
2. Definir el dominio: Definir el modelo del dominio que se utilizará en el micro servicio. Incluya las entidades del dominio, como User, Token, EmailConfirmation, etc. Además, defina las reglas de negocio que se deben seguir.
3. Definir la capa de aplicación: La capa de aplicación contiene los casos de uso y se encarga de orquestar la interacción entre el controlador y el dominio. Esta capa también se encarga de convertir los datos de la solicitud HTTP en objetos de dominio.
4. Definir la capa de infraestructura: La capa de infraestructura se encarga de la implementación concreta de las abstracciones definidas en la capa de aplicación, incluyendo la base de datos, el servicio de correo electrónico, la autenticación y la autorización. Esta capa también es responsable de la implementación del controlador.
5. Configurar la autenticación: Configurar el middleware de autenticación de JWT para que el servicio pueda validar las solicitudes entrantes.
6. Configurar la autorización: Configurar el middleware de autorización para que el servicio pueda verificar si los usuarios tienen permisos para acceder a los recursos.
7. Configurar el servicio de correo electrónico: Configurar el servicio de correo electrónico para enviar correos electrónicos a los usuarios.
8. Implementar el caso de uso de registro de usuario: Implementar el caso de uso de registro de usuario que crea una entidad de usuario y envía un correo electrónico de confirmación.
9. Implementar la confirmación de correo electrónico: Implementar el caso de uso de confirmación de correo electrónico que verifica la existencia del correo electrónico y establece la bandera de confirmación en el usuario.
10. Implementar el inicio de sesión: Implementar el caso de uso de inicio de sesión que autentica al usuario y devuelve un token JWT.
11. Implementar la recuperación de contraseña: Implementar el caso de uso de recuperación de contraseña que genera y envía un correo electrónico de recuperación de contraseña.
12. Implementar el cambio de contraseña: Implementar el caso de uso de cambio de contraseña que cambia la contraseña del usuario y envía un correo electrónico de confirmación.
13. Implementar las pruebas unitarias y de integración: Implementar pruebas unitarias y de integración para cada uno de los casos de uso.
14. Implementar la documentación: Implementar la documentación de la API utilizando Swagger.

## 1. Definición de los casos de uso

#### Caso de uso 1: Registro de usuario

Eventos:

1. Usuario crea una cuenta.
2. Usuario ingresa sus datos personales.
3. Usuario ingresa su dirección de correo electrónico.
4. Usuario ingresa su contraseña.
5. Usuario presiona el botón "Registrarse".

Comandos:

* Crear usuario.

Validaciones:

* Verificar que el correo electrónico no esté en uso.
* Verificar que la contraseña sea segura.

Respuestas:

* Usuario registrado exitosamente.
* Error al registrar el usuario.

**Caso de uso 2: Confirmación de correo electrónico**

Eventos:

* Usuario hace clic en el enlace de confirmación enviado por correo electrónico.

Comandos:

* Confirmar correo electrónico.

Validaciones:

* Verificar que el correo electrónico no haya sido confirmado previamente.

Respuestas:

* Correo electrónico confirmado exitosamente.
* Error al confirmar el correo electrónico.

**Caso de uso 3: Inicio de sesión de usuario**

Eventos:

1. Usuario ingresa su dirección de correo electrónico.
2. Usuario ingresa su contraseña.
3. Usuario presiona el botón "Iniciar sesión".

Comandos:

* Autenticar usuario.

Validaciones:

* Verificar que el correo electrónico y la contraseña sean válidos.

Respuestas:

* Usuario autenticado exitosamente.
* Error al autenticar el usuario.

**Caso de uso 4: Recuperación de contraseña**

Eventos:

1. Usuario hace clic en el enlace "¿Olvidó su contraseña?".
2. Usuario ingresa su dirección de correo electrónico.
3. Usuario presiona el botón "Enviar correo electrónico".

Comandos:

* Enviar correo electrónico de recuperación de contraseña.

Validaciones:

* Verificar que el correo electrónico sea válido.

Respuestas:

* Correo electrónico de recuperación de contraseña enviado exitosamente.
* Error al enviar el correo electrónico de recuperación de contraseña.

## 2. Definición del dominio

### Caso de uso 1: Registro de usuario

En este caso de uso, el agregado raíz es el Usuario, el cual tiene las siguientes propiedades:

* Id: identificador único del usuario.
* Nombre: nombre del usuario.
* Apellido: apellido del usuario.
* Email: dirección de correo electrónico del usuario.
* Password: contraseña del usuario.

Evento de dominio:

* UsuarioRegistrado: Se produce cuando un usuario completa el proceso de registro con éxito.

Posibles acciones adicionales (consecuencias):

* UsuarioRegistrado: Se puede enviar un correo electrónico de confirmación al usuario para verificar su dirección de correo electrónico. También se puede almacenar la información del usuario en una base de datos y generar un token de autenticación para el usuario.

El registro de usuario se realiza a través del método "RegistrarUsuario" del agregado Usuario. Este método se encarga de verificar que el correo electrónico no esté en uso y que la contraseña sea segura. En caso de que los datos sean correctos, el usuario es creado y almacenado en la base de datos.

### Caso de uso 2: Confirmación de correo electrónico

En este caso de uso, el agregado raíz es también el Usuario. El correo electrónico confirmado es una propiedad de la entidad Usuario. Al confirmar el correo electrónico, se utiliza el método.

Evento de dominio:

* CorreoElectronicoConfirmado: Se produce cuando un usuario confirma su correo electrónico con éxito.

Posibles acciones adicionales (consecuencias):

* CorreoElectronicoConfirmado: Se puede actualizar la información del usuario en la base de datos para marcar su correo electrónico como confirmado. También se puede enviar un mensaje de bienvenida al usuario.

"ConfirmarCorreoElectronico" del agregado Usuario. Este método verifica que el correo electrónico no haya sido confirmado previamente.

### Caso de uso 3: Inicio de sesión de usuario

En este caso de uso, el agregado raíz es nuevamente el Usuario. La autenticación se realiza a través del método "AutenticarUsuario" del agregado Usuario. Este método verifica que el correo electrónico y la contraseña sean válidos.

Evento de dominio:

* UsuarioIniciadoSesion: Se produce cuando un usuario inicia sesión correctamente en el sistema.
* IntentoInicioSesionFallido: Se produce cuando un usuario intenta iniciar sesión, pero las credenciales son incorrectas o el usuario no está registrado en el sistema.

Posibles acciones adicionales (consecuencias):

* UsuarioIniciadoSesion: Se puede generar un token de autenticación para el usuario y devolverlo como parte de la respuesta HTTP. También se puede registrar la hora de inicio de sesión del usuario.
* IntentoInicioSesionFallido: Se puede registrar el intento fallido de inicio de sesión en un registro de auditoría para fines de seguridad. También se puede restringir el número de intentos de inicio de sesión fallidos antes de bloquear temporalmente la cuenta del usuario.

### Caso de uso 4: Recuperación de contraseña

En este caso de uso, el agregado raíz es el CorreoElectronico, que tiene las siguientes propiedades:

* Id: identificador único del correo electrónico.
* UsuarioId: identificador del usuario al que pertenece el correo electrónico.

El correo electrónico de recuperación de contraseña se envía utilizando el método "EnviarCorreoElectronicoRecuperacionPassword" del agregado CorreoElectronico. Este método verifica que el correo electrónico sea válido y envía un correo electrónico al usuario con un enlace para recuperar la contraseña.

Evento de dominio:

* CorreoElectronicoEnviado: Se produce cuando se envía un correo electrónico con éxito al usuario para recuperar su contraseña.
* ContraseñaRecuperada: Se produce cuando un usuario completa con éxito el proceso de recuperación de contraseña.

Posibles acciones adicionales (consecuencias):

* CorreoElectronicoEnviado: Se puede enviar un correo electrónico al usuario con un enlace que lo llevará a una página de restablecimiento de contraseña. También se puede registrar la hora en que se envió el correo electrónico para fines de auditoría.
* ContraseñaRecuperada: Se puede actualizar la contraseña del usuario en la base de datos y enviar un correo electrónico de confirmación al usuario. También se puede registrar la hora en que se recuperó la contraseña para fines de auditoría.