

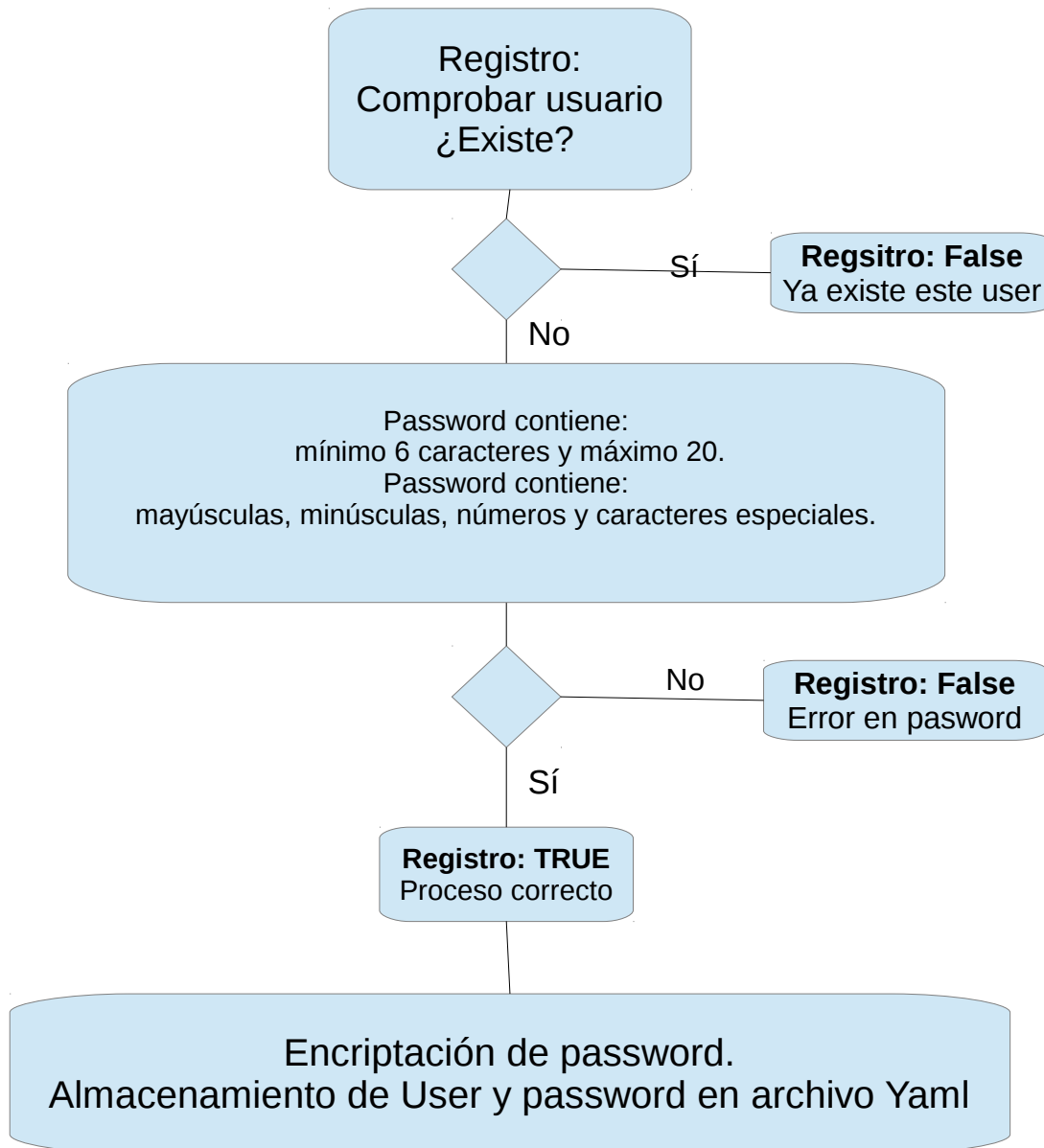
Clase: User

Clase encargada de comprobar los registros y permisos.

Métodos:

- **init** : Lista usuarios y Permisos.
- **CheckRegistro(nameUser):**
Comprobar si esta registrado
Return True / False
- **CheckPermisos(nameUser,PermisoNecesario):**
Comprobar si tiene permisos
Return True / False

Zulassung: Clase - Registro



Clase: Registro
Clase encargada de registrar usuarios.

Métodos:

- **Registro(nameUser,password):**
 - ExistUser: Comprobar si existe usuario
 - PasswordCorrect: Comprobar si es correcto el password
 - Return Registro:True / False
- **Almacena(nameUser,password):**
 - Graba en archivo yaml el user y password encriptado
- **Encrypt(password):**
 - Encripta password.
 - Return Password_Encriptado

Zulassung: Clase - Autorizador

Clase: Autorizador

Clase encargada de creación de permisos, modificación de usuarios (nombres, password, permiso), listado de usuarios(nombre). Listar Roles

Métodos:

- **init:**

Comprobar que tiene permiso ROLE_ADMIN

- **AddPermiso(Permiso):**

Comprobar si existe el permiso
Registrar Permiso

- **EditNameUser(NameUser):**

Self.CheckUser(NameUser) : Comprueba que exista el user.
Modifica NameUser.

- **EditPassword(NameUser,NewPassword):**

Self.CheckUser(NameUser) : Comprueba que exista el user.
Modifica Password del User.

- **EditPermiso(NameUser):**

Self.CheckUser(NameUser) : Comprueba que exista el user.
Modifica Permiso del User.

- **CheckUser(NameUser):**

Comprobar existencia de user.
Return True/False

-**ListUsers():**

Listar usuarios, lista según criterios Nombre , Rol

-**ListRoles():**

Listado de roles existentes en el sistema.