

# Seguridad

# Seguridad

---

Symfony tiene un componente de seguridad propio, basado en la idea de usuarios y roles.

Nos libera de:

- Comprobar si un usuario mete bien sus credenciales.
- Mantener la sesión iniciada.
- Proteger las páginas a usuarios no autenticados.
- Etc.



# Seguridad: el firewall

---

Symfony emplea elementos llamados **firewalls**:

- `/config/packages/security.yaml`
- Cada ruta de nuestra aplicación tiene uno asociado.
- Que establece:
  - Qué usuarios hay (de una tabla de la base de datos, LDAP, un fichero, etc.)
  - Cómo autenticarlos (un formulario, certificados digitales, etc.)
  - Si hace falta estar autenticado o no para acceder a la ruta.



# Seguridad: el firewall

---

Symfony emplea elementos llamados **firewalls**:

- `/config/packages/security.yaml`
- Cada ruta de nuestra aplicación tiene uno asociado.
- Que establece:
  - Qué usuarios hay (de una tabla de la base de datos, LDAP, un fichero, etc.)
  - Cómo autenticarlos (un formulario, certificados digitales, etc.)
  - Si hace falta estar autenticado o no para acceder a la ruta.

Nosotros: una entidad de Doctrine, un formulario de login y, a priori, todas se podrán ver sin estar logueado.



# Seguridad: pasos

La manera completa: <https://symfony.com/doc/current/security.html>

**Paso 1:** crear una entidad que implemente `UserInterface` y `PasswordAuthenticatedUserInterface`

`medico.sql`  
`Medico.php`

**Paso 2:** crear un formulario de login.

`LoginController.php`  
`login.html.twig`

**Paso 3:** modificar `security.yaml` para que haya un firewall que utilice:

- Un proveedor de usuarios → con la entidad creada.
- Formulario de login → los que hemos creado.
- Indicar el logout.

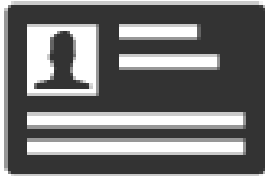
`/config/packages/  
security.yaml`

**Paso 4:** proteger los controladores o clases controladoras que queramos.

`EjemploAcceso.php`  
`acceso.html.twig`



# Seguridad: resumen

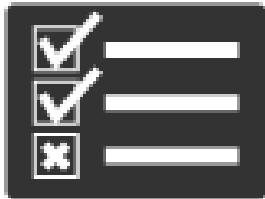


## Authentication

Who you are

### ¿Cómo autenticamos?

- Con formulario de login.
- En base a una entidad/tabla de la BBDD.
- De lo contrario → usuario anónimo.
- Una plantilla accede al usuario autenticado con `{{ app.user }}`
- Un controlador accede al usuario autenticado con `$this->getUser()`



## Authorization

What you can do

### ¿Cómo autorizamos?

- Todo permitido.
- Un controlador o clase controladora puede prohibir con `$this->denyAccessUnlessGranted();`



