

Actividad Desarrollo en Servidor.

En esta actividad vamos a desarrollar la parte de seguridad usando los paquetes de Symfony

1. Instalar los paquetes necesarios

Asegúrate de haber instalado el paquete `symfony/maker-bundle` y el `symfony/security-bundle`

```
composer require symfony/maker-bundle --dev  
composer require symfony/security-bundle
```

2. Crear una entidad User

Si aún no tienes una entidad User, puedes crearla usando el Maker Bundle:

```
php bin/console make:user
```

3. Crear el formulario de registro

Ejecuta el siguiente comando para generar un nuevo formulario:

```
php bin/console make:form
```

Asigna un nombre al formulario, por ejemplo, `RegistrationFormType`, y selecciona la entidad User como la clase de datos. Esto generará una clase básica para el formulario.

4. Crear el controlador

Genera un controlador para manejar el proceso de registro:

```
php bin/console make:controller RegistrationController
```

5. Actualizar el controlador

Modifica el controlador `RegistrationController` para manejar el formulario, guardar al usuario en la base de datos y codificar la contraseña del usuario. Aquí tienes un ejemplo:

```
namespace App\Controller;
```

```
use App\Entity\User;  
use App\Form\RegistrationFormType;  
use Doctrine\ORM\EntityManagerInterface;
```

```
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\PasswordHasher\Hasher\UserPasswordHasherInterface;
use Symfony\Component\Routing\Annotation\Route;

class RegistroController extends AbstractController
{
    #[Route('/register', name: 'app_register')]
    public function register(
        Request $request,
        UserPasswordHasherInterface $passwordHasher,
        EntityManagerInterface $entityManager
    ): Response {
        $user = new User();

        $form = $this->createForm(RegistrationFormType::class, $user);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            // Hash the plain password
            $user->setPassword(
                $passwordHasher->hashPassword(
                    $user,
                    $form->get('password')->getData()
                )
            );

            $user->setRoles($form->get('roles')->getData());

            $entityManager->persist($user);
            $entityManager->flush();
        }
    }
}
```

```

        return $this->redirectToRoute('app_music');
    }

    return $this->render('registration/register.html.twig', [
        'registrationForm' => $form->createView(),
    ]);
}
}

```

6. Actualizar la clase del formulario

En la clase `RegistrationFormType`, agrega campos como `email`, `plainPassword` y cualquier otro campo necesario. Por ejemplo:

```

namespace App\Form;

use App\Entity\User;
use Symfony\Component\Form\AbstractType;
use Symfony\Component\Form\Extension\Core\Type\PasswordType;
use Symfony\Component\Form\Extension\Core\Type\RepeatedType;
use Symfony\Component\Form\FormBuilderInterface;
use Symfony\Component\OptionsResolver\OptionsResolver;

class RegistrationFormType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options): void
    {
        $builder
            ->add('email')
            ->add('roles', ChoiceType::class, [
                'choices' => [
                    'Admin' => 'ROLE_ADMIN',

```

```

        'User' => 'ROLE_USER',

        ],

        'multiple' => true, // Permitir múltiples opciones

        'expanded' => true, ])// Renderiza como checkboxes

        //->add('password') quitar este y sustituir por este

        ->add('password', RepeatedType::class, [

            'type' => PasswordType::class,

            'first_options' => ['label' => 'Password'],

            'second_options' => ['label' => 'Repeat Password'],

        ]);

    };

}

public function configureOptions(OptionsResolver $resolver): void
{
    $resolver->setDefaults([

        'data_class' => User::class,

    ]);
}
}

```

7. Crear la plantilla Twig

Crea un archivo `register.html.twig` en la carpeta `templates/registration/` e incluye el código para renderizar el formulario:

```

{% extends 'base.html.twig' %}

{% block body %}

    <h1>Registro</h1>

    {{ form_start(registrationForm) }}

        {{ form_widget(registrationForm) }}

    {{ form_end(registrationForm) }}

{% endblock %}

```

```
<button class="btn">Registrar</button>

{{ form_end(registrationForm) }}

{% endblock %}
```