

Desarrollamos
{ talento tech }



GeeksHubs
academy _



RoadMap

enlace

<https://symfony.com/roadmap>

<https://symfony.com/roadmap>





Paco Gomez Arnal

Youtuber, educador y creador de contenidos. Desarrollador de aplicaciones y entusiasta de la enseñanza y la tecnología



<https://www.linkedin.com/in/paco-gomez-arnal/>



<http://www.youtube.com/c/PacoGomez>



<http://www.repositoriocompartido.es>

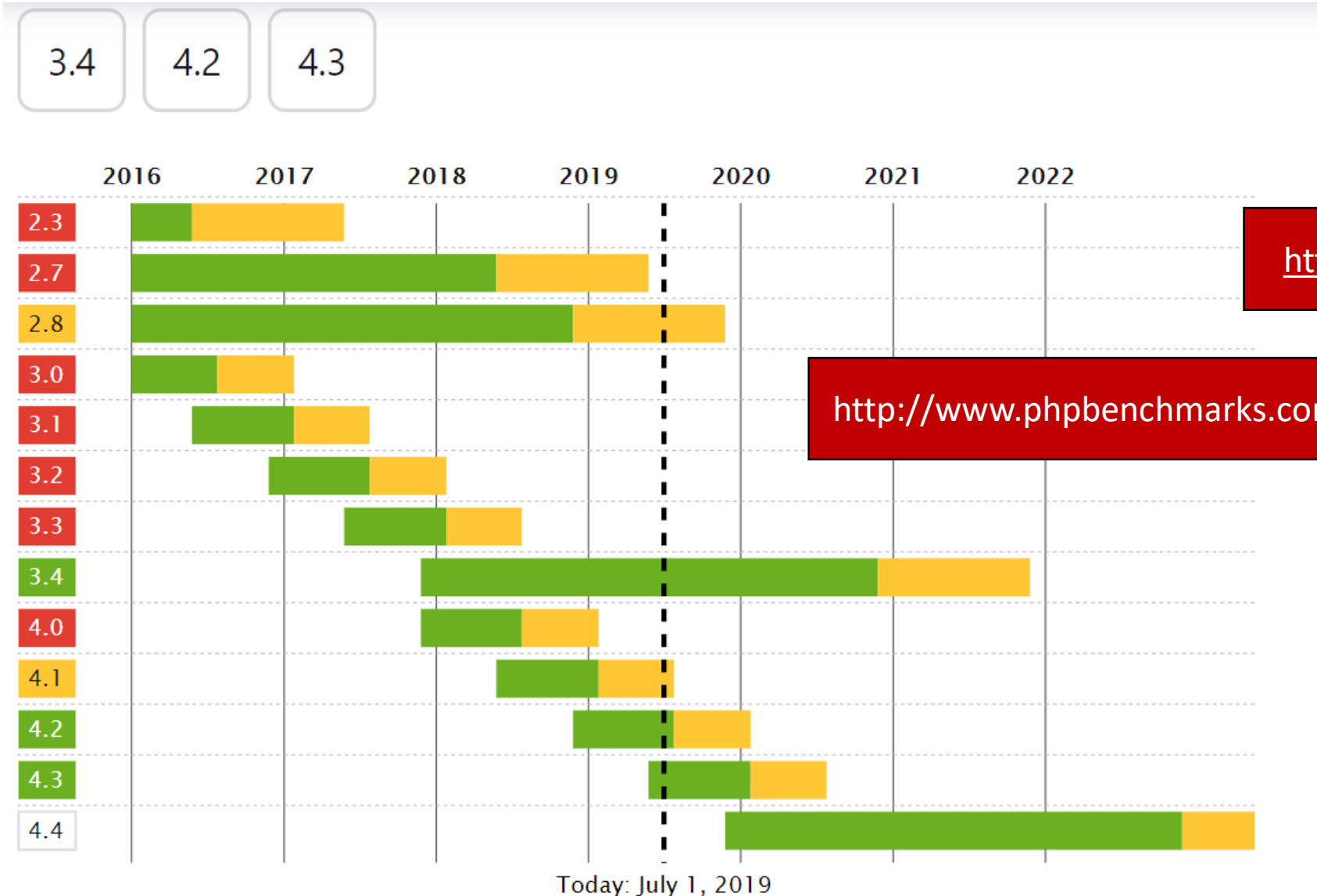


[@pacogomezarnal](https://twitter.com/pacogomezarnal)

fjgomez@florida-uni.es

paco.gomez.arnal@gmail.com

RoadMap



<https://symfony.com/roadmap>

<http://www.phpbenchmarks.com/en/comparator/framework>

Inicio

```
$ composer create-project symfony/skeleton clubBasket  
  
$ cd clubBasket\  
  
$ php -S 127.0.0.1:8000 -t public  
//Lanza el servidor de php Zend sobre la carpeta visible public  
  
$ php bin\console server:start  
//NO FUNCIONA!!!!!!  
  
$ php bin\console about  
$ php bin\console
```

Inicio

```
$ git init  
$ git add  
$ ...
```



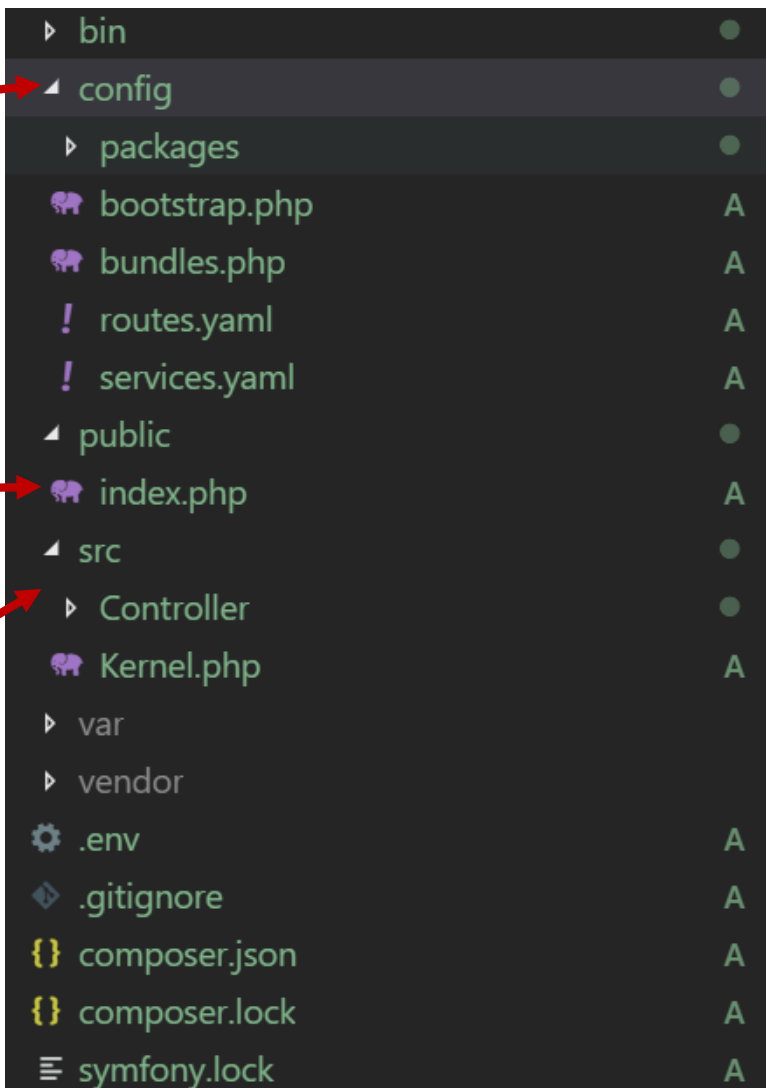
<https://symfony.com/doc/current/setup.html>

Estructura

CONFIG

CONTROLLER

CODIGO



Server

```
$ composer require server  
//Instala el server bundle  
//Actualiza composer  
//Actualiza console  
  
$ php bin/console server:run
```


Rutas y anotaciones

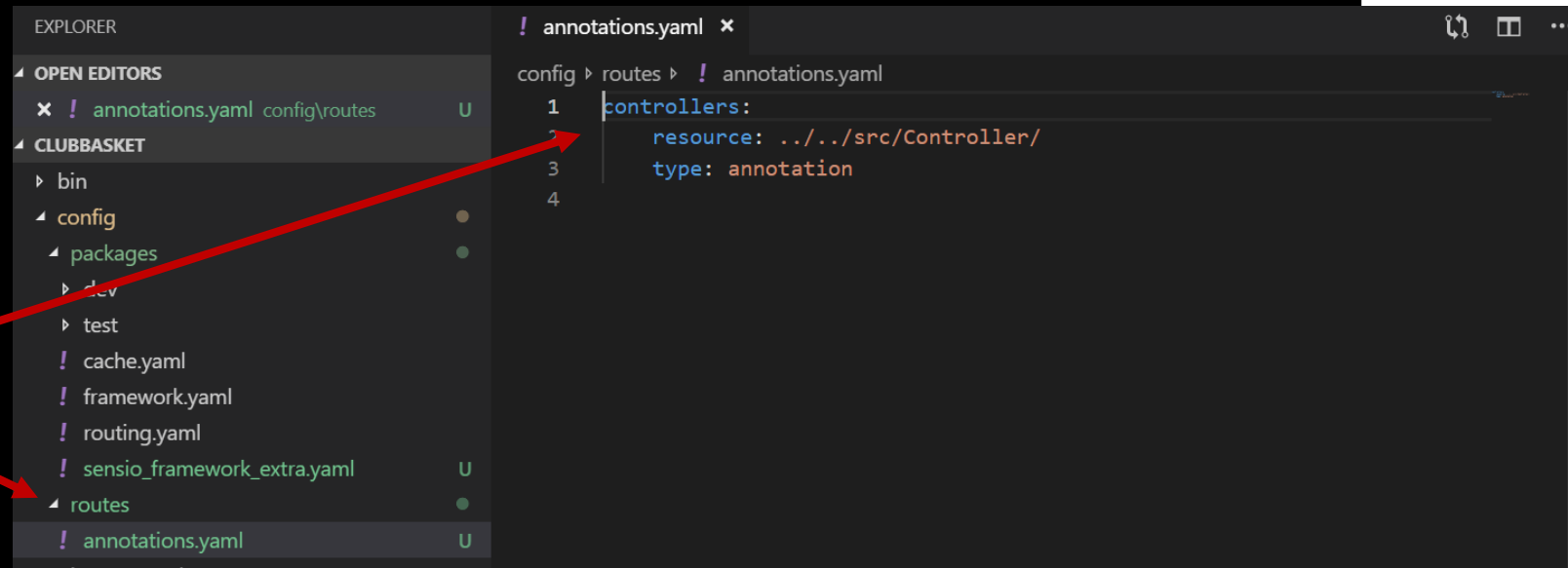
\$ composer require annotations

//Instala el server bundle

//Actualiza composer

//Actualiza console

CONFIG



<https://symfony.com/doc/current/routing.html>

Primera pagina

```
<?php
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Component\HttpFoundation\Response;

class HomeController extends AbstractController
{
    /**
     * @Route("/", name="homepage")
     */
    public function homepage()
    {
        return new Response('Mi primera página con Symfony 4');
    }
}
```

Segunda ruta

```
/**
 * @Route("/detalle/{id}", name="detalle")
 */
public function detalle($id=3)
{
    return new Response("Detalle de la página $id");
}
```

Flex

FLEX

<https://flex.symfony.com>

Symfony Recipes Server

Search for recipes

Symfony Flex is the way to manage Symfony applications.

It is based on **Symfony Recipes**, which are a set of automated instructions to integrate third-party packages into Symfony applications.

This page lists these great building blocks for your Symfony applications.

Learn More

- Using recipes in your Symfony applications
- Create a recipe for a public package
- Source code for official recipes, community Flex itself

All Symfony Recipes available

ad3n/ratchet-bundle

contrib

Package details Recipe

adback/adback-sdk-php-symfony

contrib

Package details Recipe

ajardin/docker-symfony

contrib

Package details Recipe

algolia/algolia-se

contrib

Package details

```
composer.json ▸ ...
1 {
2     "type": "project",
3     "license": "proprietary",
4     "require": {
5         "php": "^7.1.3",
6         "ext-ctype": "*",
7         "ext-iconv": "*",
8         "sensio/framework-extra-bundle": "^5.3",
9         "symfony/console": "4.3.*",
10        "symfony/dotenv": "4.3.*",
11        "symfony/flex": "^1.3.1",
12        "symfony/framework-bundle": "4.3.*",
13        "symfony/web-server-bundle": "4.
14        "symfony/yaml": "4.3.*"
15    },
16    "require-dev": {
17    },
18    "config": {
19        "preferred-inst
20        ".*": "dist"
21    },
22    },
23    "sort-packages"
24    },
25    "autoload": {
26        "psr-4": {
```

ALIAS

<https://flex.symfony.com/>

<https://symfony.com/doc/current/setup/flex.html>

symfony/web-server-bundle

official

Aliases server web-server

web-server-bundle

Package details Recipe

Twig

```
$ composer require twig  
//Instala el twig bundle
```


symfony/twig-bundle

official

Aliases `template` `templates` `twig`
`twig-bundle`

[Package details](#) [Recipe](#)

 bundles.php x

config ▸  bundles.php

```
1 <?php  
2  
3 return [  
4     Symfony\Bundle\FrameworkBundle\FrameworkBundle::class => ['all' => true],  
5     Symfony\Bundle\WebServerBundle\WebServerBundle::class => ['dev' => true],  
6     Sensio\Bundle\FrameworkExtraBundle\SensioFrameworkExtraBundle::class => ['all' => true],  
7     Symfony\Bundle\TwigBundle\TwigBundle::class => ['all' => true],  
8 ];  
9
```

! twig.yaml x

config ▸ packages ▸ ! twig.yaml

```
1 twig:  
2     default_path: '%kernel.project_dir%/templates'  
3     debug: '%kernel.debug%'  
4     strict_variables: '%kernel.debug%'  
5
```

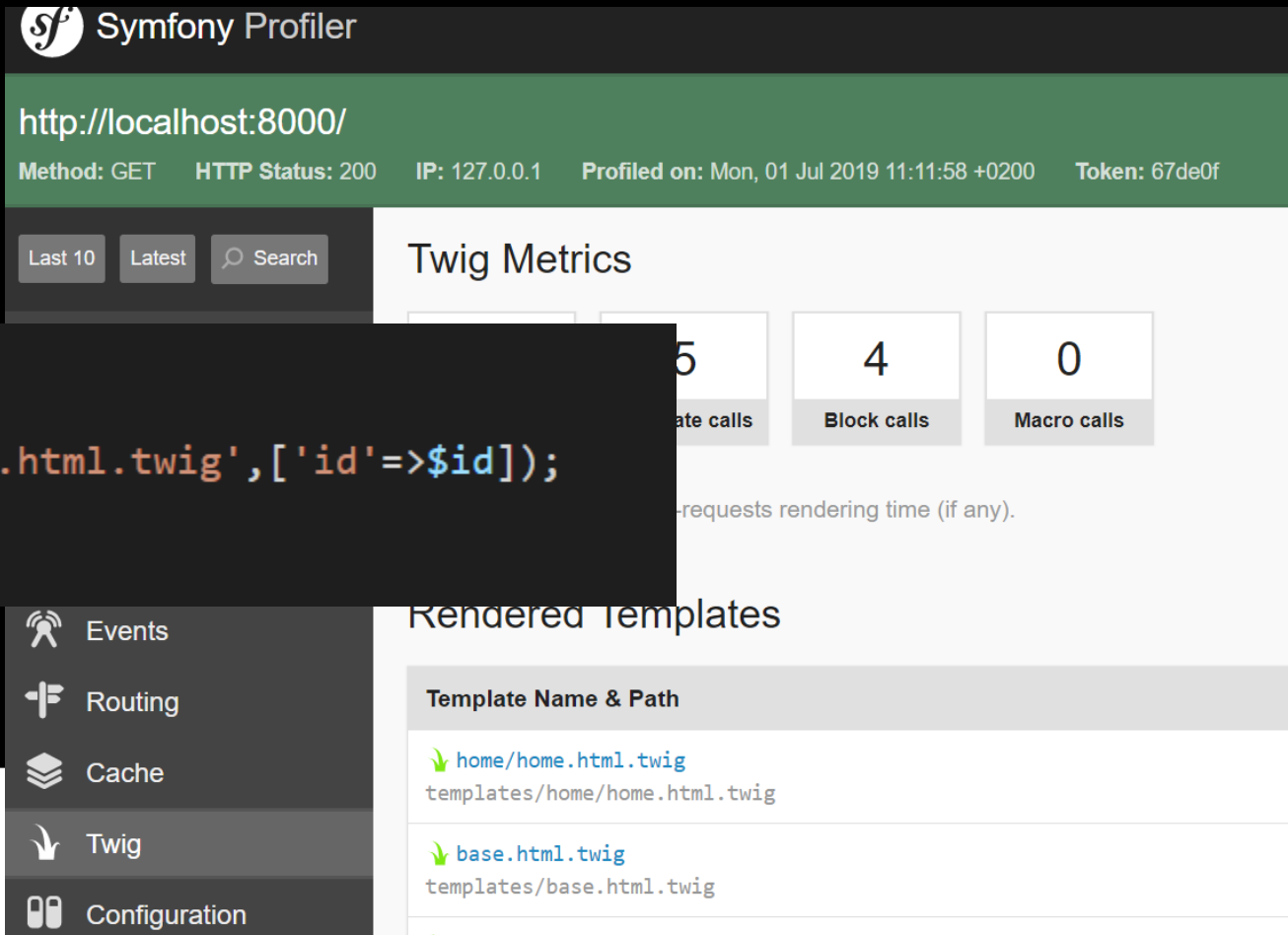
<https://symfony.com/doc/current/templating.html>

Web Debug ToolBar

```
$ composer require profiler
```

```
$ composer require debug
```

```
$ composer unpack debug
```



Symfony Profiler

http://localhost:8000/

Method: GET HTTP Status: 200 IP: 127.0.0.1 Profiled on: Mon, 01 Jul 2019 11:11:58 +0200 Token: 67de0f

Last 10 Latest Search

Twig Metrics

Category	Count
Block calls	5
Block calls	4
Macro calls	0

requests rendering time (if any).

Rendered Templates

Template Name & Path
home/home.html.twig templates/home/home.html.twig
base.html.twig templates/base.html.twig

- Events
- Routing
- Cache
- Twig
- Configuration

```
{
    dump($id,$this);
    return $this->render('home/detalle.html.twig',['id'=>$id]);
}
```

```
//Ruta de prueba con Response
```

```
{% BLOCK body %}
<div>
    {{dump()}}
    {{id}}
</div>
{% endblock %}
```

Asset

```
$ composer require asset  
//Instala el asset bundle
```



<https://symfony.com/doc/current/components/asset.html>

Doctrine

ENV

```
$ composer require symfony/orm-pack  
$ composer require symfony/maker-bundle  
//Instala ORM y Doctrine
```

```
# Configure your db driver and server_version in config/packages/doctrine.yaml  
DATABASE_URL=mysql://root@127.0.0.1:3306/clubBasket
```

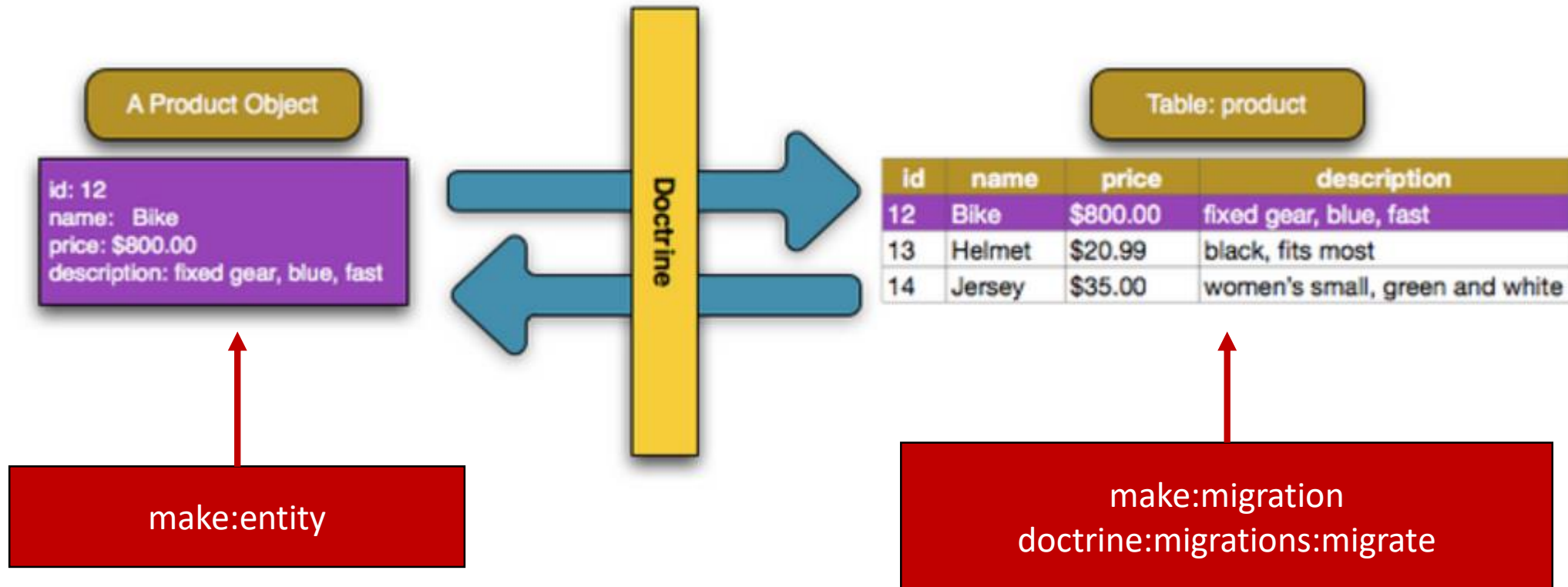
Database Configuration

- * Modify your DATABASE_URL config in `.env`
- * Configure the `driver` (mysql) and `server_version` (5.7) in `config/packages/doctrine.yaml`

▶ vendor
⚙ .env
📄 .gitignore
📄 composer.json

<https://symfony.com/doc/current/doctrine.html>

Doctrine



<https://symfony.com/doc/current/doctrine.html>

Doctrine Relaciones

```
/**
 * @ORM\OneToMany(targetEntity="App\Entity\Resultado", mappedBy="equipolocal")
 */
private $resultadoslocales;

/**
 * @ORM\OneToMany(targetEntity="App\Entity\Resultado", mappedBy="equipovisitante")
 */
private $resultadosvisitantes;
```

Form

```
$ composer require symfony/form  
$ composer require --dev symfony/maker-bundle  
//Instala ORM y Doctrine
```

```
# Configure your db driver and server_version in config/packages/doctrine.yaml  
DATABASE_URL=mysql://root@127.0.0.1:3306/clubBasket
```



<https://symfony.com/doc/current/forms.html>

Form AbstractType

```
<?php
namespace App\Form;

use Symfony\Component\Form\AbstractType;
use Symfony\Component\Form\Extension\Core\Type\SubmitType;
use Symfony\Component\Form\FormBuilderInterface;

class EquipoType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('categoria')
            ->add('sexo')
            ->add('numjugadores')
            ->add('save', SubmitType::class, ['label' => 'Crear Equipo']);
    }
}
```

Form y Doctrine

```
/**
 * @Route("/nuevo", name="nuevoEquipo")
 */
public function nuevoEquipo(Request $request)
{
    //Creamos un nuevo objeto de tipo equipo
    $equipo = new Equipo();
    //Creamos el nuevo formulario
    $form = $this->createForm(EquipoType::class, $equipo);

    //Recogemos la información del POST
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $equipo = $form->getData();
        $entityManager = $this->getDoctrine()->getManager();
        $entityManager->persist($equipo);
        $entityManager->flush();
        return $this->redirectToRoute('home');
    }
    return $this->render('equipo/nuevoEquipo.html.twig', [
        'form' => $form->createView(),
    ]);
}
```

Twig y Doctrine

Lista de equipos

#	Categoria	Sexo	Numero jugadores	Editar
1	benjamin	mixto	13	
2	benjamin tecnificado	masculino	13	
3	alevin	femenino	15	

Nuevo Equipo



Form y Doctrine. Actualizar

```
/**
 * @Route("/equipo")
 */
class EquipoController extends AbstractController
{
    /**
     * @Route("/nuevo/{id}", name="nuevoEquipo")
     */
    public function nuevoEquipo(Request $request, $id=null)
    {
        if($id)
        {
            //Buscamos un objeto de tipo equipo
            $repository = $this->getDoctrine()->getRepository(Equipo::class);
            $equipo = $repository->find($id);
            $saveText = "Actualiza";
        }else{
            //Creamos un nuevo objeto de tipo equipo
            $equipo = new Equipo();
            $saveText = "Nuevo";
        }
        //Creamos el nuevo formulario
        $form = $this->createForm(EquipoType::class, $equipo, array(['saveText'=>$saveText])
```

Form ConfigureOptions

```
    ;  
    }  
    public function configureOptions(OptionsResolver $resolver)  
    {  
        $resolver->setRequired(array(  
            'saveText'  
        ));  
    }  
}
```


Delete Doctrine

```
/**
 * @Route("/borrar/{id}", name="borrarEquipo")
 */
public function borrarEquipo($id=null)
{
    if($id)
    {
        //Buscamos un objeto de tipo equipo
        $repository = $this->getDoctrine()->getRepository(Equipo::class);
        $equipo = $repository->find($id);
        $entityManager = $this->getDoctrine()->getManager();
        $entityManager->remove($equipo);
        $entityManager->flush();
        return $this->redirectToRoute('listaEquipos');
    }else{
        //No se borra nada
        return $this->redirectToRoute('listaEquipos');
    }
}
```

Console make:crud

```
$ php bin/console  
//Todas las opciones
```

```
$ composer require validator security-csrf  
//Instala librerías necesarias par realizar CRUD
```

```
$ php bin/console make:crud  
//Crea CRUD sobre Entidad
```

```
C:\symfony\clubBasket (master -> origin)
```

```
λ php bin/console make:crud
```

```
The class name of the entity to create CRUD (e.g. TinyKangaroo):
```

```
> Resultado
```

```
Resultado
```

```
created: src/Controller/ResultadoController.php
```

```
created: src/Form/ResultadoType.php
```

```
created: templates/resultado/_delete_form.html.twig
```

```
created: templates/resultado/_form.html.twig
```

```
created: templates/resultado/edit.html.twig
```

```
created: templates/resultado/index.html.twig
```

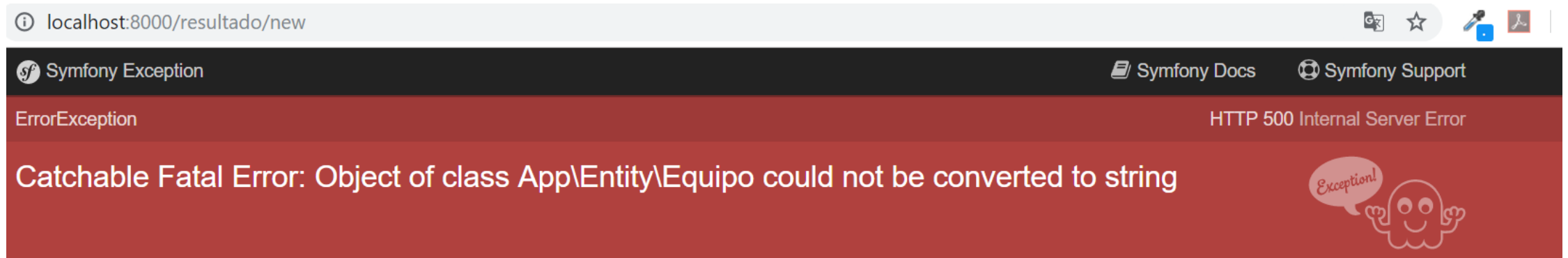
```
created: templates/resultado/new.html.twig
```

```
created: templates/resultado/show.html.twig
```

```
Success!
```

```
Next: Check your new CRUD by going to /resultado/
```

Console make:crud



Hacking Fields

Text Fields ¶

- [TextType](#)
- [TextareaType](#)
- [EmailType](#)
- [IntegerType](#)
- [MoneyType](#)
- [NumberType](#)
- [PasswordType](#)
- [PercentType](#)
- [SearchType](#)
- [UrlType](#)
- [RangeType](#)
- [TelType](#)
- [ColorType](#)

Choice Fields ¶

- [ChoiceType](#)
- [EntityType](#)
- [CountryType](#)
- [LanguageType](#)
- [LocaleType](#)
- [TimezoneType](#)
- [CurrencyType](#)

Date and Time Fields

- [DateType](#)
- [DateIntervalType](#)
- [DateTimeType](#)
- [TimeType](#)
- [BirthdayType](#)

Other Fields ¶

- [CheckboxType](#)
- [FileType](#)
- [RadioType](#)

Field Groups ¶

- [CollectionType](#)
- [RepeatedType](#)

Hidden Fields ¶

- [HiddenType](#)

Buttons ¶

- [ButtonType](#)
- [ResetType](#)
- [SubmitType](#)

Base Fields ¶

- [FormType](#)

Hacking Fields

```
$builder  
    ->add('puntoslocal',NumberType::class,  
        ['label'=>'Puntos Local']  
    )  
    ->add('puntosvisitante',NumberType::class,  
        ['label'=>'Puntos Local'])
```

```
$builder  
    ->add('puntoslocal',NumberType::class,  
        ['label'=>'Puntos Local']  
    )  
    ->add('puntosvisitante',NumberType::class,  
        ['label'=>'Puntos Local'])
```

Hacking Fields

```
public function buildForm(FormBuilderInterface $builder, array $options)
{
    $builder
        ->add('categoria')
        ->add('sexo', ChoiceType::class, [
            'choices' => [
                'Mixto' => true,
                'Masculino' => false,
                'Femenino' => false
            ]
        ])
        ->add('numjugadores')
        ->add('save', SubmitType::class, ['label' => $options['saveText']])
    ;
}
```

Rendering forms

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

{% block title %}New Resultado{% endblock %}

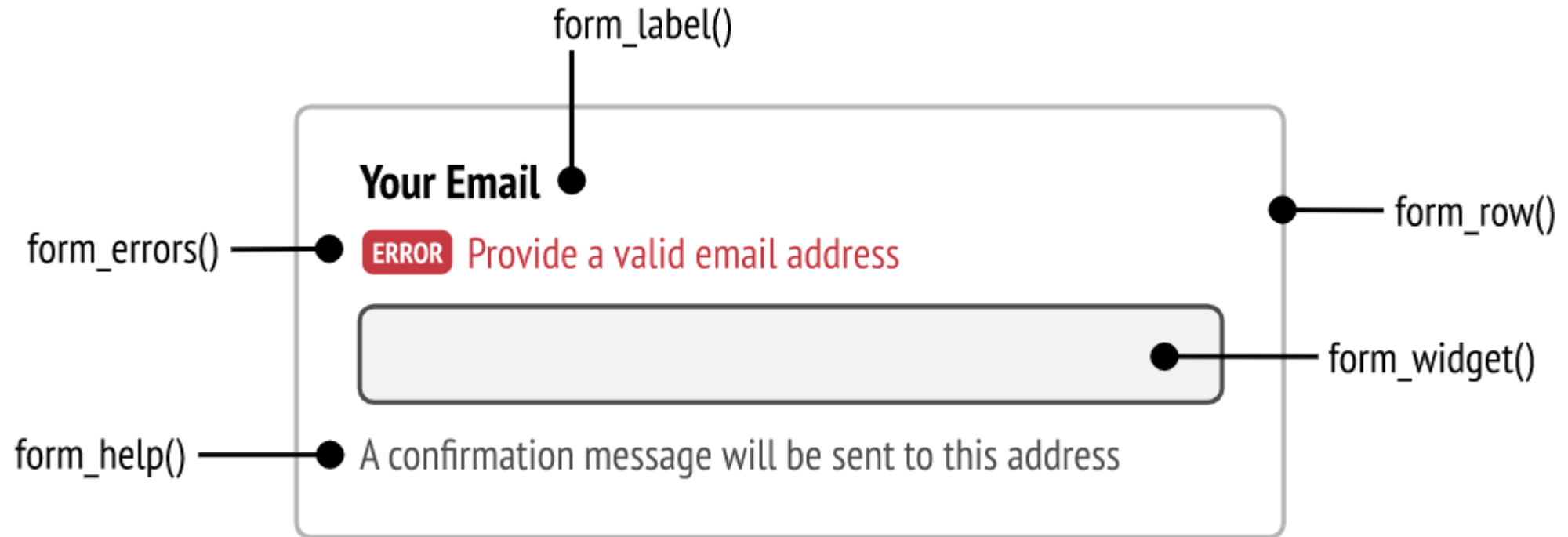
{% block body %}
    <h1>Create new Resultado</h1>

    {{ include('resultado/_form.html.twig') }}

    <a href="{ { path('resultado_index') } }">back to list</a>
{% endblock %}
```

INCLUDE

Rendering forms



Rendering forms

```
{% block body %}
  <div class="form_small">
    {{ form_start(form) }}
    <div>
      {{ form_errors(form) }}
    </div>
    <div class="form-group">
      {{ form_label(form.categoria) }}
      {{ form_widget(form.categoria, {'attr': {'class': 'form-control'}}) }}
      {{ form_help(form.categoria, {'attr': {'class': 'form-text text-muted'}}) }}
    </div>
    <div class="form-group">
      {{ form_label(form.sexo) }}
      {{ form_widget(form.sexo, {'attr': {'class': 'form-control'}}) }}
      {{ form_help(form.sexo, {'attr': {'class': 'form-text text-muted'}}) }}
    </div>
    <div class="form-group">
      {{ form_label(form.numjugadores) }}
      {{ form_widget(form.numjugadores, {'attr': {'class': 'form-control'}}) }}
      {{ form_help(form.numjugadores, {'attr': {'class': 'form-text text-muted'}}) }}
    </div>
    {{ form_label(form.save) }}
    {{ form_end(form) }}
  </div>
{% endblock %}
```

https://symfony.com/doc/current/form/form_customization.html#reference-forms-twig-form

Rendering forms. Themes

```
1  {% extends 'base.html.twig' %}
2
3  {% form_theme form _self %}
4
5  {# this overrides the widget of any field of type integer, but only in the
6     forms rendered inside this template #}
7  {% block integer_widget %}
8      <div class="...">
9          {# ... render the HTML element to display this field ... #}
10     </div>
11 {% endblock %}
12
13 {# this overrides the entire row of the field whose "id" = "product_stock" (and whose
14     "name" = "product[stock]") but only in the forms rendered inside this template #}
15 {% block _product_stock_row %}
16     <div class="..." id="...">
17         {# ... render the entire field contents, including its errors ... #}
18     </div>
19 {% endblock %}
20
21 {# ... render the form ... #}
```

https://symfony.com/doc/current/form/form_themes.html

Symfony security

```
$ composer require symfony/security-bundle
```

//Instala el Bundle de Seguridad

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

AYUDANTE

```
created: src/Entity/Usuario.php  
created: src/Repository/UsuarioRepository.php  
updated: src/Entity/Usuario.php  
updated: config/packages/security.yaml
```

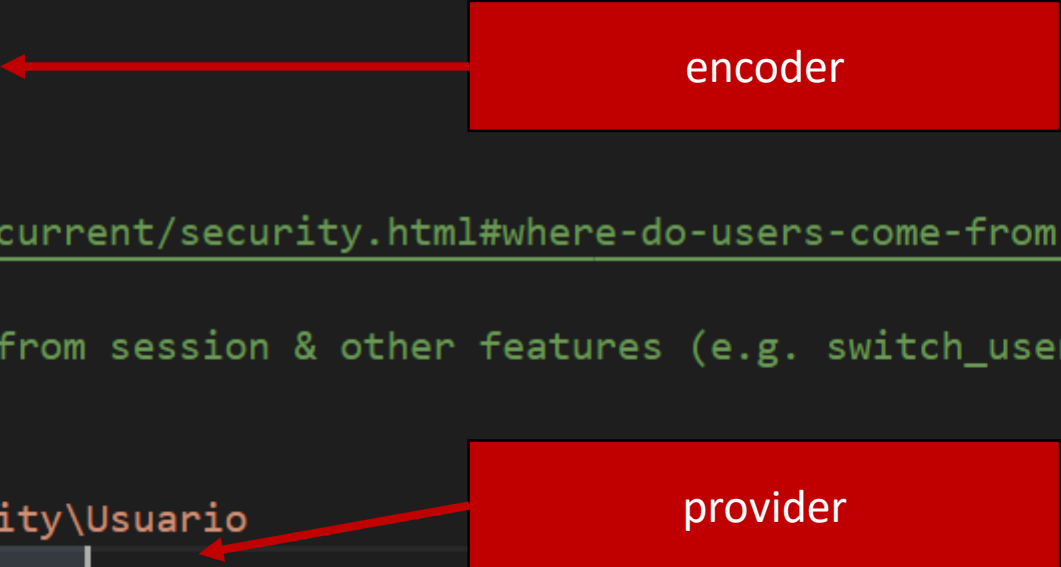
```
$ php bin/console make:migration
```

```
$ php bin/console doctrine:migrations:migrate
```

Symfony security, password

```
security:
  encoders:
    App\Entity\Usuario:
      algorithm: argon2i

# https://symfony.com/doc/current/security.html#where-do-users-come-from-user-providers
providers:
  # used to reload user from session & other features (e.g. switch_user)
  app_user_provider:
    entity:
      class: App\Entity\Usuario
      property: username
```



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

Symfony security, registro

```
class UsuarioType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('email', EmailType::class)
            ->add('plainPassword', RepeatedType::class, array(
                'type' => PasswordType::class,
                'first_options' => array('label' => 'Contraseña'),
                'second_options' => array('label' => 'Repetir Contraseña'),
            ))
            ->add('registrar', SubmitType::class, array('label' => "Registrar"));
    }
}
```

RepeatedType

Symfony security, registro

```
/**
 * @Route("/registro/", name="registro")
 */
public function registro(Request $request, UserPasswordEncoderInterface $passwordEncoder)
{
    $usuario = new Usuario();
    //Construyendo el formulario
    $form = $this->createForm(UsuarioType::class,$usuario);
    $form->handleRequest($request);
    if ($form->isSubmitted() && $form->isValid()) {
        // 3) Encode the password (you could also do this via Doctrine listener)
        $password = $passwordEncoder->encodePassword($usuario, $usuario->getPlainPassword());
        $usuario->setPassword($password);

        // 3c) $roles
        $usuario->setRoles(array('ROLE_USER'));

        // 4) save the User!
        $entityManager = $this->getDoctrine()->getManager();
        $entityManager->persist($usuario);
        $entityManager->flush();

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

    // replace this example code with whatever you need
    return $this->render('home/registro.html.twig',array('form'=>$form->createView(),'titulo'=>'Nu
}
```

https://symfony.com/doc/current/doctrine/registration_form.html

Symfony security, firewall

```
firewalls:
  dev:
    pattern: ^/(_(profiler|wdt)|css|images|js)/
    security: false
  main:
    anonymous: true
    provider: app_user_provider
    form_login:
      login_path: login
      check_path: login
    logout:
      path: /logout/
      target: /

  # activate different ways to authenticate
  # https://symfony.com/doc/current/security.html#firewalls-authentication

  # https://symfony.com/doc/current/security/impersonating\_user.html
  # switch_user: true

# Easy way to control access for large sections of your site
# Note: Only the *first* access control that matches will be used
access_control:
  # - { path: ^/admin, roles: ROLE_ADMIN }
  - { path: ^/equipo, roles: ROLE_USER }
```

FIREWALL

REGLA

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

Implementar login

```
firewalls:
  dev:
    pattern: ^/(_(profiler|wdt)|css|images|js)/
    security: false
  main:
    anonymous: true
    provider: app_user_provider
    form_login:
      login_path: login
      check_path: login
    logout:
      path: /logout/
      target: /

# activate different ways to authenticate
# https://symfony.com/doc/current/security.html#firewalls-authentication

# https://symfony.com/doc/current/security/impersonating\_user.html
# switch_user: true

# Easy way to control access for large sections of your site
# Note: Only the *first* access control that matches will be used
access_control:
  # - { path: ^/admin, roles: ROLE_ADMIN }
  - { path: ^/equipo, roles: ROLE_USER }
```

FIREWALL

REGLA

https://symfony.com/doc/current/security/form_login_setup.html

Implementar login

```
{% extends 'base.html.twig' %}
{% block body %}
<div class="form_small">
  <form method="post">
    {% if error %}
      <div class="alert alert-danger">{{ error.messageKey|trans(error.message) }}</div>
    {% endif %}
    <div class="form-group">
      <label for="username" class="sr-only">Username</label>
      <input type="text" value="{{ last_username }}" name="username" id="username" class="form-control">
    </div>
    <div class="form-group">
      <label for="password" class="sr-only">Password</label>
      <input type="password" name="password" id="password" class="form-control" placeholder="Password">
    </div>
    <input type="hidden" name="_csrf_token"
      value="{{ csrf_token('authenticate') }}"
    >
    <button class="btn btn-lg btn-primary" type="submit">
      Entrar
    </button>
  </form>
</div>
{% endblock %}
```

USERNAME

PASSWORD

Parametros de configuración

```
main:
  anonymous: true
  provider: app_user_provider
  form_login:
    username_parameter : username
    password_parameter : password
    login_path: login
    check_path: login
  logout:
    path: /logout/
    target: /
```

Implementar logout

```
config ▸ ! routes.yaml
1  #index:
2  #   path: /
3  #   controller: App\Controller\
4  logout:
5  |   path: /logout/
```

EXTRA 1: Repositorio

```
class ResultadoRepository extends ServiceEntityRepository
{
    public function __construct(RegistryInterface $registry)
    {
        parent::__construct($registry, Resultado::class);
    }

    public function lastTenResults()
    {
        return $this->createQueryBuilder('r')
            ->orderBy('r.id', 'DESC')
            ->setMaxResults(10)
            ->getQuery()
            ->getResult();
    }
}
```

ALIAS

FUNCIONES

RESULTADO

EXTRA 2: Validation

\$ composer require symfony/validator doctrine/annotations

CONFIGURACION

```
# config/packages/framework.yaml
framework:
    validation: { enabled: true }
```

```
/**
 * @ORM\Column(type="string", length=255)
 * @Assert\NotBlank
 */
private $categoria;
```

USO

<https://symfony.com/doc/current/validation.html>

EXTRA 3: Enviroment & Deploy

```
$ php bin/console cache:clear
```

```
15  
16 ###> symfony/framework-bundle ###  
17 APP_ENV=prod  
18 APP_SECRET=51e657007b28e2720864e0b26022026b
```

<https://symfony.com/doc/current/configuration.html>
<https://symfony.com/doc/current/deployment.html>