

src/Controller/PokemonsController.php

```
<?php
```

```
namespace App\Controller;

use App\Entity\Categorias;
use App\Entity\Pokemons;
use App\Repository\PokemonsRepository;
use Doctrine\ORM\EntityManager;
use Doctrine\ORM\EntityManagerInterface;
use Doctrine\Persistence\ManagerRegistry;
use Symfony\Bridge\Doctrine\Form\Type\EntityType;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\Form\Extension\Core\Type\ChoiceType;
use Symfony\Component\Form\Extension\Core\Type\IntegerType;
use Symfony\Component\Form\Extension\Core\Type\NumberType;
use Symfony\Component\Form\Extension\Core\Type\SubmitType;
use Symfony\Component\Form\Extension\Core\Type\TextType;
use Symfony\Component\HttpFoundation\JsonResponse;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;
use Symfony\Component\Validator\Constraints\Json;
```

```
#[Route('/pokemons', name: 'app_pokemons_')]
class PokemonsController extends AbstractController
{
    // Plantilla para métodos
    #[Route('/index', name: 'index')]
    public function index(): Response
    {
        return $this->render('pokemons/index.html.twig', [
            'controller_name' => 'PokemonsController',
        ]);
    }

    // 04. Inserción 1 registro con parámetros y FK
    #[Route(
        '/insertar/{cat}/{nombre}/{altura}/{peso}/{sexo}',
        name: 'insertarParams'
    )]
    public function insertarParams(
        int $cat,
        string $nombre,
        int $altura,
        float $peso,
        bool $sexo,
        ManagerRegistry $doctrine
    ): Response {
        $gestorEntidades = $doctrine->getManager();

        $pokemon = new Pokemons;
        $pokemon->setNombre($nombre);
        $pokemon->setAltura($altura);
        $pokemon->setPeso($peso);
        $pokemon->setSexo($sexo);

        // Gentileza Juan Carlos ;)
```

```

// Para claves foráneas, se introduce el objeto ENTERO
$categoria = new Categorias();
// Obtengo el repo de categorias
$repoCategorias =
    $gestorEntidades->getRepository(Categorias::class);
// Saco el objeto categoria completo
$categoria = $repoCategorias->find($cat);
$pokemon->setIdCategoria($categoria);

$gestorEntidades->persist($pokemon);
$gestorEntidades->flush();

return new Response("Pokemon insertado con ID: "
    . $pokemon->getId());
}

// 05. Consulta completa (findAll)
#[Route('/verPokemons', name: 'verpokemons')]
public function verPokemons(EntityManagerInterface $gestorEntidades): Response
{
    $repoPokemons = $gestorEntidades->getRepository(Pokemons::class);
    $pokemons = $repoPokemons->findAll();

    return $this->render('pokemons/index.html.twig', [
        'controller_name' => 'PokemonsController',
        'Bichos' => $pokemons,
    ]);
}

// 06. Consulta completa, salida array JSON
#[Route('/verPokemonsJSON', name: 'verpokemonsjson')]
public function verPokemonsJSON(PokemonsRepository $repoPokemons): Response
{
    $pokemons = $repoPokemons->findAll();
    $datos = [];
    foreach ($pokemons as $pokemon) {
        $datos[] = [
            "idPokemon" => $pokemon->getId(),
            "nombre" => $pokemon->getNombre(),
            "altura" => $pokemon->getAltura(),
            "peso" => $pokemon->getPeso(),
            "sexo" => $pokemon->isSexo(),
            "categoria" => $pokemon->getIdCategoria()->getCategoria()
        ];
    }

    return new JsonResponse($datos);
}

// 07. Consulta por parámetro, salida array JSON
#[Route('/verPokemonsJSON/{sexo}', name: 'verpokemonsjsonparam')]
public function verPokemonsJSONParam(PokemonsRepository $repoPokemons, bool $sexo):
Response
{
    // Consultar pokemons por sexo
    $pokemons = $repoPokemons->findBy(["sexo" => $sexo,]);

    $datos = [];
    foreach ($pokemons as $pokemon) {

```

```

        $datos[] = [
            "idPokemon" => $pokemon->getId(),
            "nombre" => $pokemon->getNombre(),
            "altura" => $pokemon->getAltura(),
            "peso" => $pokemon->getPeso(),
            "sexo" => $pokemon->isSexo(),
            "categoria" => $pokemon->getIdCategoria()->getCategoria()
        ];
    }

    //return new JsonResponse($datos);
    return $this->json($datos);
}

// 08. Consulta por parámetro y ordenación, salida array JSON
#[Route(
    '/verPokemonsOrdenadosJSON/{sexo}',
    name: 'verpokemonsOrdenadosjsonparam'
)]
public function verPokemonsOrdenadosJSONParam(PokemonsRepository $repoPokemons, bool $sexo): Response
{
    // Consultar pokemons por sexo
    // Y ordenarlos por altura de mayor a menor
    $pokemons =
        $repoPokemons->findBy(
            ["sexo" => $sexo,],
            ["altura" => "DESC"]
        );

    $datos = [];
    foreach ($pokemons as $pokemon) {
        $datos[] = [
            "idPokemon" => $pokemon->getId(),
            "nombre" => $pokemon->getNombre(),
            "altura" => $pokemon->getAltura(),
            "peso" => $pokemon->getPeso(),
            "sexo" => $pokemon->isSexo(),
            "categoria" => $pokemon->getIdCategoria()->getCategoria()
        ];
    }

    //return new JsonResponse($datos);
    return $this->json($datos);
}

// 09. Actualizar con parámetros
#[Route(
    '/actualizar/{id}/{altura}/{peso}',
    name: 'actualizarparams'
)]
public function actualizarParams(
    ManagerRegistry $doctrine,
    int $id,
    int $altura,
    float $peso
): Response {
    $gestorEntidades = $doctrine->getManager();

```

```

// Sacamos el pokemon que vamos a actualizar
$repoPokemons = $gestorEntidades->getRepository(Pokemons::class);
$pokemon = $repoPokemons->find($id);

if (!$pokemon) {
    throw $this->createNotFoundException("Pokemon NO encontrado");
}

$pokemon->setAltura($altura);
$pokemon->setPeso($peso);
$gestorEntidades->flush();

// Vamos a hacer una redirección
return $this->redirectToRoute("app_pokemons_verpokemons");
}

// 10. Eliminación con parámetro (id)
#[Route('/eliminar/{id}', name: 'eliminar')]
public function eliminar(
    EntityManagerInterface $gestorEntidades,
    int $id
): Response {
    // Sacamos el pokemon que vamos a eliminar
    $repoPokemons = $gestorEntidades->getRepository(Pokemons::class);
    $pokemon = $repoPokemons->find($id);

    // Borro y actualizo
    $gestorEntidades->remove($pokemon);
    $gestorEntidades->flush();
    return new Response("Pokemon eliminado con ID: " . $id);
}

// 11. Formulario
// 2 inyecciones: la solicitud (request) y el doctrine
#[Route('/formulario', name: 'formulario')]
public function formulario(Request $request,
ManagerRegistry $doctrine): Response
{
    // a. Creamos el objeto a guardar vacío
    $pokemon = new Pokemons();

    // b. Creamos el objeto formulario
    $formulario = $this->createFormBuilder($pokemon)
        ->add("nombre", TextType::class,
        ["attr" => ["class" => "form-control"] ])
        ->add("altura", IntegerType::class,
        ["attr" => ["class" => "form-control"] ])
        ->add("peso", NumberType::class,
        ['attr' => [ 'class' => 'form-control',
        'step' => '0.01' ],
        'html5' => true ])

        ->add("sexo", ChoiceType::class,
        [ "choices" => [
            "Hembra" => true,
            "Macho" => false, ],
        "attr" => ["class" => "form-control"]
        ])

    // Añadimos el campo del FK

```

```
->add("idCategoria", EntityType::class,
[
    "class" => Categorias::class,    // Entidad
    // Choice label -> nombre tabla FK en minusculas
    "choice_label" => "categoria",
    "placeholder" => "Selecciona categoria",
    "attr" => ["class" => "form-select"]
])

->add("guardar", SubmitType::class,
    ["attr" => ["class" => "btn btn-danger"],
     "label" => "Insertar Pokémon"])
->getForm();

// c. Tratar el formulario
// Recoger datos del formulario e insertar en la BBDD
$formulario->handleRequest($request);

if ($formulario->isSubmitted() && $formulario->isValid()) {
    // Guardamos el pokemon
    $gestorEntidades = $doctrine->getManager();
    $gestorEntidades->persist($pokemon);
    $gestorEntidades->flush();

    // Redireccionamos
    return $this->redirectToRoute("app_pokemons_verpokemons");
}

// d. Pintar el formulario
return $this->render('pokemons/formulario.html.twig', [
    'controller_name' => 'PokemonsController',
    "formulario" => $formulario,
]);
}
}
```