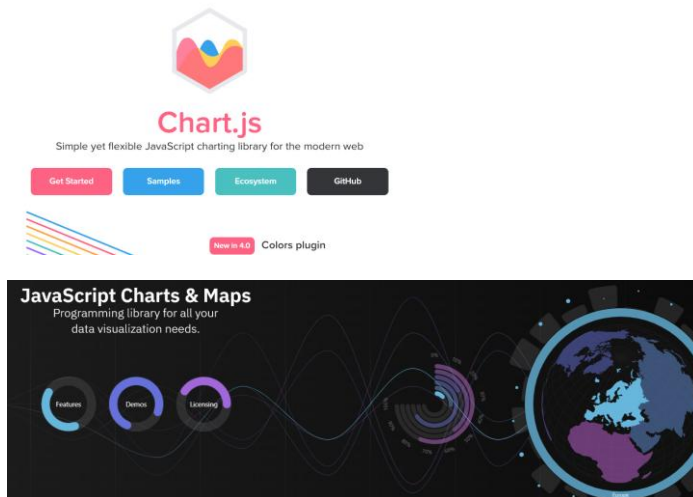


Desarrollo en Servidor

Estudio de los casos de uso del usuario Manager

Debes seleccionar en primer lugar una librería JS que te ayude a dibujar las gráficas



El siguiente controller es un ejemplo de como se puede implementar los casos de uso del manager

```
<?php

namespace App\Controller;

use App\Repository\PlaylistCancionRepository;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\JsonResponse;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class EstadisticasController extends AbstractController
{
    #[Route('/estadisticas', name: 'estadisticas')]
    public function index(PlaylistCancionRepository
$playlistCancionRepository): Response
    {
        // Obtener datos de reproducciones por playlist
```

```

        $datos = $playlistCancionRepository-
>obtenerReproduccionesPorPlaylist();

        return $this->render('estadisticas/index.html.twig', [
            'datos' => $datos
        ]);
    }

    #[Route('/estadisticas/datos', name: 'estadisticas_datos')]
    public function obtenerDatos(PlaylistCancionRepository
$playlistCancionRepository): JsonResponse
    {
        $datos = $playlistCancionRepository-
>obtenerReproduccionesPorPlaylist();

        return $this->json($datos); // convierte el array $datos en una
respuesta JSON.
    }
}

```

La respuesta de este controller sería por ejemplo

```

[
    {"playlist": "Rock Clásico", "totalReproducciones": 120},
    {"playlist": "Pop ", "totalReproducciones": 80},
    {"playlist": "Jazz ", "totalReproducciones": 45}
]

```

Detalle del PlaylistCancionRepository. Fíjate que por cada playlist se agrupa y se aplica una función de agregación

```

<?php

namespace App\Repository;

use App\Entity\PlaylistCancion;
use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;

```

```

use Doctrine\Persistence\ManagerRegistry;

class PlaylistCancionRepository extends ServiceEntityRepository
{
    public function __construct(ManagerRegistry $registry)
    {
        parent::__construct($registry, PlaylistCancion::class);
    }

    public function obtenerReproduccionesPorPlaylist(): array
    {
        return $this->createQueryBuilder('pc')
            ->select('p.nombre AS playlist, SUM(pc.reproducciones) AS
totalReproducciones')
            ->join('pc.playlist', 'p')
            ->groupBy('p.id')
            ->getQuery()
            ->getResult();
    }
}

```

Por último vamos a fijarnos en como se usa la librería de java script para mostrar un grafico

```

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

{% block title %}Estadísticas{% endblock %}

{% block body %}
    <div class="container">
        <h2>Estadísticas de Reproducciones por Playlist</h2>
        <canvas id="reproduccionesChart"></canvas>
    </div>

    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script>

```

```

<!-- se ejecuta cuando la página ha terminado de cargarse

document.addEventListener("DOMContentLoaded", function () {

    <!--hace una solicitud HTTP GET a la URL que devuelve la ruta generada por Symfony
(path('estadisticas_datos')).-->

    fetch("{ path('estadisticas_datos') }")

        .then(response => response.json())

        .then(data => {

            const labels = data.map(item => item.playlist);

            const values = data.map(item => item.totalReproducciones);


            const ctx = document.getElementById('reproduccionesChart').getContext('2d');

            new Chart(ctx, {

                type: 'bar',

                data: {

                    labels: labels,

                    datasets: [{

                        label: 'Total Reproducciones',

                        data: values,

                        backgroundColor: 'rgba(54, 162, 235, 0.5)',

                        borderColor: 'rgba(54, 162, 235, 1)',

                        borderWidth: 1

                    }]

                },

                options: {

                    responsive: true,

                    scales: {

                        y: {

                            beginAtZero: true

                        }

                    }

                }

            });

        });

    });

</script>
{% endblock %}

```