



FRAMEWORKS BACKEND: COMPARACIÓN

RIAH



21 DE OCTUBRE DE 2024

GRUPO AIR

Manuel Perales Briones

0. CONCEPTOS BÁSICOS

Necesitamos encontrar un Framework de trabajo que dé funcionalidad al Frontend que desarrollemos. Por ello, analizaremos distintas herramientas según su nivel de adecuación, mediante las siguientes estimaciones:

- Accesibilidad: qué tan sencillo es acceder a las funcionalidades de la herramienta (tarifas, disponibilidad al público, etc.).
- Relevancia: a qué nivel nos proporciona las herramientas necesarias.
- Compatibilidad: cómo se integra en el Stack global: herramientas compatibles.
- Usabilidad: qué tanta eficiencia nos da el Framework, tanto para aprender como para trabajar.
- Experiencia: nivel del desarrollador en dicha herramienta (las herramientas que no he utilizado previamente tienen la puntuación mínima).

MUY MAL
No tiene ninguno de
los parámetros
deseados

MAL
Tiene muy pocos
parámetros deseados

NO ESTÁ MAL
Los parámetros más
importantes los cumple

BIEN
Cumple casi todos los
parámetros deseados

MUY BIEN
Cumple todos los
parámetros requeridos



1. LISTA

NodeJS

<https://nodejs.org/docs/latest/api/>

- Accesibilidad

Gratuito para todo el público.

- Relevancia
 - Permite comunicación en tiempo real.
 - Permite utilizar módulos que accedan a las BBDDs que necesitemos.
 - Permite conexiones seguras a Frontend mediante HTTPS.

- Compatibilidad
 - Podemos conectarnos a nuestros SGDBs (en nuestro caso, MySQL y MongoDB).
 - Podemos conectarnos a las principales tecnologías de Frontend (Angular, React, VueJS).

- Usabilidad
 - Documentación legible
 - Creación sencilla de proyectos.
 - Configuración de servidor sencilla.
 - Incorporación de módulos sencilla.
 - Lenguaje simple y flexible (JavaScript).

- Experiencia

Uso hasta la fecha muy limitado, únicamente creando proyectos React. Experiencia reciente respecto al lenguaje de programación.

Django

<https://docs.djangoproject.com/en/5.1/>

- Accesibilidad

Gratuito para todo el público. Al ser un paquete de Python con un “-pip install” es suficiente.

- Relevancia

- Se basa en el patrón MVC, recomendable para este proyecto.
- Al trabajar con Python, se pueden utilizar distribuciones para las BBDDs que necesitemos utilizar.
- Permite comunicación en tiempo real.
- Permite conexiones seguras a Frontend mediante HTTPS.

- Compatibilidad

- Podemos conectarnos a nuestros SGDBs (en nuestro caso, MySQL y MongoDB).
- Podemos conectarnos a las principales tecnologías de Frontend (Angular, React, VueJS).

- Usabilidad

- Documentación legible.
- Incorpora tutorial sencillo.
- Lenguaje de muy alto nivel (Python).
- Basado en el patrón principal (MVC).
- Fichero de configuración sencillo.

- Experiencia

Nula experiencia con el framework. Experiencia en diversos proyectos respecto al lenguaje de programación.

Spring

<https://spring.io/>

- Accesibilidad

Gratuito para todo el público. Se trata de una dependencia de Maven (java).

- Relevancia

- Al trabajar con Java, se pueden utilizar JPARepository, que permite conectarse mediante un fichero de configuración a muchas BBDDs.
- Permite comunicación en tiempo real abriendo un puerto de escucha.
- Permite conexiones seguras a Frontend mediante HTTPS.

- Compatibilidad

- Podemos conectarnos a nuestros SGDBs (en nuestro caso, MySQL y MongoDB).
- Podemos conectarnos a las principales tecnologías de Frontend (Angular, React, VueJS).

- Usabilidad

- Documentación legible.
- Notaciones excesivas y de difícil memorización (@Transient, @Entity, JPARepository, etc.).
- Ficheros de configuración sencillos.
- Manejo sencillo de conexión HTTP.

- Experiencia

Mucha experiencia con el Framework (se ha desarrollado ya un proyecto web con él previamente). Mucha experiencia con el lenguaje de programación (Java).

Laravel

<https://laravel.com/docs/11.x>

- Accesibilidad

Gratuito para todo el público. Simplemente hay que crear un proyecto utilizando un terminal.

- Relevancia

- Se basa en el patrón MVC, recomendable para este proyecto.
- Integra por defecto la BBDD relacional.
- Permite comunicación en tiempo real.
- Integra ya funciones como la autenticación.

- Compatibilidad

- Podemos conectarnos a nuestros SGDBs (en nuestro caso, MySQL y MongoDB).
- Podemos conectarnos a las principales tecnologías de Frontend (Angular, React).

- Usabilidad

- Documentación legible.
- Lenguaje de programación muy maduro (PHP) pero no tan intuitivo como otras alternativas modernas.
- Basado en el patrón principal (MVC).
- Fichero de configuración complejo.

- Experiencia

Nula experiencia con el Framework. Nula experiencia con el lenguaje de programación.

Phoenix

<https://hexdocs.pm/phoenix/Phoenix.html>

- Accesibilidad

Gratuito para todo el público, para ello se requiere tener Exilir y Erlang.



- Relevancia
 - Integra por defecto la BBDD relacional, la cual es configurable al SGDB deseado.
 - Permite manejar eventos y ejecuta en tiempo real (el lenguaje está diseñado para ello, además de soportar grandes cargas).



- Compatibilidad
 - Podemos conectarnos a nuestros SGDBs (en nuestro caso, MySQL y MongoDB).
 - Aunque tiene conexión directa con React, no tiene compatibilidad nativa con otros Frameworks como Angular.



- Usabilidad
 - Documentación legible.
 - Lenguaje de alto nivel y permite funciones (Elixir).
 - Fichero de configuración accesible.
 - Manejo de comandos sencillos.



- Experiencia

Nula experiencia con el Framework. Nula experiencia con el lenguaje de programación.



2. CONCLUSIONES

	Acces.	Relev.	Compat.	Usab.	Exp.
NodeJS					
Django					
Spring					
Laravel					
Phoenix					

Aunque todos los Frameworks brindan funcionalidades útiles para el proyecto, **Spring** es la mejor opción ya que, aunque no es tan usable como otros, brinda las funcionalidades necesarias y, gracias a la experiencia previa, permitirá una eficiencia en el desarrollo.