

UNIVERSIDAD TECNICA PARTICULAR DE LOJA

TECNOLOGIAS WEB

[ACDB1-35%] Foro: Investigar y analizar tecnologías de BackEnd para el desarrollo de aplicaciones web, ventajas y beneficios de usarlas

NOMBRE: KEVIN ANGELO PORTILLA NARVAEZ

CLASE: 100-ECTS-RED



Las tecnologías de BackEnd son fundamentales para el desarrollo de aplicaciones web, ya que gestionan la lógica del servidor, las bases de datos y la comunicación entre el servidor y el cliente. Las tecnologías más destacadas en el desarrollo de BackEnd son:

1. **Node.js:** Un entorno de ejecución para JavaScript que permite ejecutar código en el servidor. Es conocido por su eficiencia y capacidad para manejar múltiples conexiones simultáneamente
2. **Python (Django):** Django es un framework de alto nivel para Python que facilita el desarrollo rápido y seguro de aplicaciones web
3. **PHP (Laravel):** PHP es un lenguaje de scripting ampliamente utilizado en el desarrollo web. Laravel es un framework popular que simplifica muchas tareas comunes en el desarrollo web
4. **Java (Spring):** Spring es un framework para Java que proporciona una infraestructura robusta para el desarrollo de aplicaciones empresariales escalables
5. **Ruby (Ruby on Rails):** Ruby on Rails es un framework que facilita el desarrollo rápido de aplicaciones web con Ruby, conocido por su simplicidad y productividad
6. **C# (.NET):** Utilizado con el framework .NET, C# es ideal para aplicaciones empresariales y de gran escala
7. **Go:** Un lenguaje desarrollado por Google, conocido por su eficiencia y rendimiento, ideal para aplicaciones que requieren alta concurrencia

Estas tecnologías tienen sus propias fortalezas y son adecuadas para diferentes tipos de proyectos.

Para la parte 1 del foro he escogido las siguientes tecnologías:

1. Node.js
2. Python (Django)
3. Java (Spring)
4. Go

Las cuales a continuación realizaremos un detalle de cada una de ellas:

1. **Node.js:**

Descripción:

Ideal si se tiene experiencia con JavaScript. Es muy eficiente y se usa mucho en aplicaciones que requieren manejar una gran cantidad de conexiones simultáneas. El framework **Express.js** es una excelente adición para simplificar el desarrollo. **Node.js** es un entorno de ejecución de JavaScript que permite ejecutar código JavaScript en el

lado del servidor. Está basado en el motor V8 de Google Chrome y es conocido por su eficiencia y capacidad para manejar múltiples conexiones simultáneamente.

Ventajas:

1. **Rendimiento y escalabilidad:** Utiliza un modelo de eventos no bloqueante, lo que permite manejar muchas conexiones simultáneamente.
2. **Facilidad de aprendizaje:** Si ya conoces JavaScript, aprender Node.js es relativamente sencillo.
3. **Ecosistema de paquetes:** NPM, el gestor de paquetes de Node.js, tiene una gran cantidad de módulos disponibles.
4. **Comunidad activa:** Hay una gran comunidad de desarrolladores que contribuyen y ofrecen soporte.
5. **Unificación del lenguaje:** Permite usar JavaScript tanto en el frontend como en el backend.

Desventajas:

1. **Programación asíncrona:** Puede ser difícil de manejar para quienes no están acostumbrados.
2. **Falta de soporte para multihilo:** Node.js utiliza un solo hilo, lo que puede limitar su rendimiento en operaciones intensivas de CPU.
3. **Madurez relativa:** Es una tecnología relativamente joven, lo que puede resultar en cambios frecuentes en la API.
4. **Manejo de errores:** La programación asíncrona puede complicar el manejo de errores.
5. **Uso intensivo de CPU:** No es ideal para aplicaciones que requieren un uso intensivo de la CPU.

Tipo de Licencia: MIT License

La MIT License es una licencia de software libre y de código abierto que se originó en el Instituto de Tecnología de Massachusetts (MIT) a finales de los años 80. Es conocida por ser una licencia permisiva, lo que significa que impone muy pocas restricciones sobre el uso, modificación y distribución del software.

Sus características Principales:

- **Permisos:** Permite a los usuarios usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar y vender copias del software.

- **Condiciones:** La única condición es que se incluya el aviso de copyright y la declaración de permiso en todas las copias o partes sustanciales del software.
- **Limitaciones:** El software se proporciona "tal cual", sin garantías de ningún tipo, expresas o implícitas, incluyendo, pero no limitándose a garantías de comerciabilidad, idoneidad para un propósito particular y no infracción.

Ventajas:

1. **Simplicidad:** Es fácil de entender y aplicar.
2. **Flexibilidad:** Permite el uso en proyectos tanto de código abierto como propietarios.
3. **Compatibilidad:** Es compatible con muchas otras licencias de software.
4. **Popularidad:** Es una de las licencias más utilizadas en proyectos de código abierto.
5. **Libertad:** Ofrece a los desarrolladores una gran libertad para usar y distribuir el software.

Desventajas:

1. **Falta de protección:** No protege contra el uso indebido del software.
2. **Sin copyleft:** No obliga a que las modificaciones también se distribuyan bajo la misma licencia.
3. **Ambigüedad en patentes:** No aborda explícitamente los derechos de patentes.
4. **Responsabilidad limitada:** No ofrece garantías ni asume responsabilidades por el software.
5. **Reutilización en software propietario:** Permite que el software se integre en proyectos propietarios sin compartir mejoras.

Referencias:

<https://mit-license.org/>

https://en.wikipedia.org/wiki/MIT_License

Curva de Aprendizaje: Es Moderada. Es más fácil si ya tienes experiencia con JavaScript, pero la programación asíncrona puede ser un desafío

Ejemplo de Uso: Desarrollo de aplicaciones en tiempo real como chats y juegos en línea

Página Oficial: <https://nodejs.org/en/>

2. Python (Django):

Descripción:

Django es un framework de alto nivel para el desarrollo web en Python, diseñado para facilitar la creación de aplicaciones web rápidas y seguras siendo un framework muy completo que te permite desarrollar aplicaciones web rápidamente. Python es fácil de aprender y tiene una gran comunidad de apoyo.

Ventajas:

1. **Productividad:** Permite desarrollar aplicaciones rápidamente gracias a sus características integradas.
2. **Seguridad:** Incluye medidas de seguridad por defecto, como protección contra inyección SQL y XSS.
3. **Escalabilidad:** Puede manejar aplicaciones de cualquier tamaño.
4. **Comunidad y soporte:** Gran comunidad y abundante documentación.
5. **Mantenibilidad:** Su estructura clara facilita el mantenimiento del código.

Desventajas:

1. **Curva de aprendizaje:** Puede ser complejo para principiantes.
2. **Rendimiento:** No es el más rápido en comparación con otros frameworks más ligeros.
3. **Flexibilidad limitada:** Su enfoque basado en convenciones puede ser restrictivo.
4. **Tamaño del paquete:** Incluye muchas funcionalidades por defecto, lo que puede ser excesivo para proyectos pequeños.
5. **Configuración inicial:** Puede requerir más configuración inicial en comparación con microframeworks como Flask.

Tipo de Licencia: BSD License.

La BSD License (Berkeley Software Distribution License) es una licencia de software libre y de código abierto que permite el uso, modificación y distribución del software con mínimas restricciones. Es conocida por ser una licencia permisiva, lo que significa que impone pocas restricciones sobre cómo se puede usar, modificar y redistribuir el software.

Características Principales:

- **Permisos:** Permite a los usuarios usar, copiar, modificar y distribuir el software, tanto en formato de código fuente como en binario.

- **Condiciones:** Requiere que se mantenga el aviso de copyright y la lista de condiciones en todas las redistribuciones del código fuente y en la documentación asociada.
- **Limitaciones:** El software se proporciona "tal cual", sin garantías de ningún tipo, incluyendo garantías implícitas de comerciabilidad o idoneidad para un propósito particular.

Tipos de Licencias BSD:

1. **BSD de 4 cláusulas (original):** Incluía una cláusula de publicidad que requería que todos los materiales publicitarios mencionaran el software original.
2. **BSD de 3 cláusulas (revisada):** Eliminó la cláusula de publicidad, haciéndola más compatible con otras licencias como la GPL.
3. **BSD de 2 cláusulas (simplificada):** Aún más simplificada, utilizada principalmente en proyectos como FreeBSD.

Ventajas:

1. **Flexibilidad:** Permite el uso en proyectos tanto de código abierto como propietarios.
2. **Compatibilidad:** Compatible con muchas otras licencias de software.
3. **Simplicidad:** Fácil de entender y aplicar.
4. **Popularidad:** Utilizada en muchos proyectos importantes de software libre.
5. **Libertad:** Ofrece a los desarrolladores una gran libertad para usar y distribuir el software.

Desventajas:

1. **Falta de protección:** No protege contra el uso indebido del software.
2. **Sin copyleft:** No obliga a que las modificaciones también se distribuyan bajo la misma licencia.
3. **Ambigüedad en patentes:** No aborda explícitamente los derechos de patentes.
4. **Responsabilidad limitada:** No ofrece garantías ni asume responsabilidades por el software.
5. **Reutilización en software propietario:** Permite que el software se integre en proyectos propietarios sin compartir mejoras.

Referencias:

https://en.wikipedia.org/wiki/BSD_licenses

<https://snyk.io/learn/what-is-bsd-license/>

Curva de Aprendizaje: Moderada a alta. Requiere tiempo para familiarizarse con su estructura y convenciones.

Ejemplo de Uso: Desarrollo de aplicaciones web complejas y API RESTful.

Referencias: <https://www.djangoproject.com/>

3. Java (Spring):

Descripción:

Spring es un framework de código abierto para el desarrollo de aplicaciones en Java. Ofrece soporte para la creación de aplicaciones empresariales brindando robustez y escalabilidad, Java con el framework Spring es una excelente opción.

Ventajas:

1. **Modularidad:** Permite usar solo los módulos necesarios.
2. **Flexibilidad:** Se adapta a diferentes arquitecturas y patrones de diseño.
3. **Desarrollo rápido:** Spring Boot facilita la configuración y el desarrollo rápido.
4. **Comunidad activa:** Gran cantidad de recursos y soporte disponibles.
5. **Integración:** Compatible con una amplia variedad de tecnologías y plataformas.

Desventajas:

1. **Curva de aprendizaje:** Puede ser complejo para principiantes.
2. **Configuración:** Requiere una configuración inicial considerable.
3. **Sobrecarga:** Puede ser excesivo para aplicaciones pequeñas.
4. **Rendimiento:** Puede no ser tan rápido como otros frameworks más ligeros.
5. **Dependencias:** Puede tener muchas dependencias, lo que complica la gestión.

Tipo de Licencia: Apache License 2.0.

La Apache License 2.0 es una licencia de software libre y de código abierto creada por la Apache Software Foundation (ASF). Es conocida por ser una licencia permisiva, lo que significa que permite el uso, modificación y distribución del software con pocas restricciones.

Características Principales:

- **Permisos:** Permite a los usuarios usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar y vender copias del software.
- **Condiciones:** Requiere que se incluya un aviso de copyright y una declaración de permiso en todas las copias o partes sustanciales del software. También requiere que se incluya un archivo de NOTICE con cualquier distribución que contenga un aviso de copyright, atribuciones, notas de cambios, etc.
- **Limitaciones:** El software se proporciona "tal cual", sin garantías de ningún tipo, expresas o implícitas, incluyendo, pero no limitándose a garantías de comerciabilidad, idoneidad para un propósito particular y no infracción.

Ventajas:

1. **Flexibilidad:** Permite el uso en proyectos tanto de código abierto como propietarios.
2. **Compatibilidad:** Compatible con muchas otras licencias de software.
3. **Protección de patentes:** Incluye una cláusula explícita de concesión de patentes, lo que protege a los usuarios de demandas por infracción de patentes.
4. **Simplicidad:** Fácil de entender y aplicar.
5. **Popularidad:** Utilizada en muchos proyectos importantes de software libre, incluyendo proyectos de la Apache Software Foundation.

Desventajas:

1. **Falta de copyleft:** No obliga a que las modificaciones también se distribuyan bajo la misma licencia.
2. **Responsabilidad limitada:** No ofrece garantías ni asume responsabilidades por el software.
3. **Ambigüedad en la atribución:** La cláusula de NOTICE puede ser interpretada de diferentes maneras, lo que puede causar confusión.
4. **Reutilización en software propietario** Permite que el software se integre en proyectos propietarios sin compartir mejoras.

5. **Complejidad legal:** Aunque es más simple que algunas licencias, puede ser más compleja que otras licencias permisivas como la MIT License.

Referencias:

<https://www.apache.org/licenses/LICENSE-2.0>

Curva de Aprendizaje: Alta. Requiere tiempo para dominar sus múltiples módulos y configuraciones.

Ejemplo de Uso: Desarrollo de aplicaciones empresariales y microservicios4.

Página Oficial: <https://spring.io/>

4. **Go:**

Descripción:

Go, también conocido como Golang, es un lenguaje de programación moderno desarrollado por Google. Es conocido por su simplicidad, eficiencia y rendimiento, Go es ideal para aplicaciones que requieren alta concurrencia, al ser desarrollado por Google está ganando popularidad rápidamente.

Ventajas:

1. **Rendimiento:** Al ser un lenguaje compilado, es muy rápido.
2. **Concurrencia:** Maneja la concurrencia de manera eficiente con goroutines.
3. **Simplicidad:** Sintaxis clara y fácil de aprender.
4. **Biblioteca estándar:** Amplia y completa, reduce la necesidad de bibliotecas externas.
5. **Seguridad:** Fuerte énfasis en la seguridad y prevención de errores.

Desventajas:

1. **Lenguaje joven:** Aún en desarrollo en comparación con otros lenguajes más establecidos.
2. **Bibliotecas:** Falta de bibliotecas maduras en algunos dominios.
3. **Gestión de memoria:** Puede requerir gestión manual en ciertos escenarios.
4. **Paradigma de programación:** Usa composición en lugar de herencia, lo que puede ser un desafío para algunos.
5. **Manejo de errores:** Usa la devolución de errores en lugar de excepciones, lo que puede ser menos intuitivo.

Tipo de Licencia: BSD License.

La **BSD License** (Berkeley Software Distribution License) es una licencia de software libre y de código abierto que permite el uso, modificación y distribución del software con mínimas restricciones. Es conocida por ser una licencia permisiva, lo que significa que impone pocas restricciones sobre cómo se puede usar, modificar y redistribuir el software.

Características Principales:

- **Permisos:** Permite a los usuarios usar, copiar, modificar y distribuir el software, tanto en formato de código fuente como en binario.
- **Condiciones:** Requiere que se mantenga el aviso de copyright y la lista de condiciones en todas las redistribuciones del código fuente y en la documentación asociada.
- **Limitaciones:** El software se proporciona "tal cual", sin garantías de ningún tipo, incluyendo garantías implícitas de comerciabilidad o idoneidad para un propósito particular.

Tipos de Licencias BSD:

1. **BSD de 4 cláusulas (original):** Incluía una cláusula de publicidad que requería que todos los materiales publicitarios mencionaran el software original.
2. **BSD de 3 cláusulas (revisada):** Eliminó la cláusula de publicidad, haciéndola más compatible con otras licencias como la GPL.
3. **BSD de 2 cláusulas (simplificada):** Aún más simplificada, utilizada principalmente en proyectos como FreeBSD.

Ventajas:

1. **Flexibilidad:** Permite el uso en proyectos tanto de código abierto como propietarios.
2. **Compatibilidad:** Compatible con muchas otras licencias de software.
3. **Simplicidad:** Fácil de entender y aplicar.
4. **Popularidad:** Utilizada en muchos proyectos importantes de software libre.
5. **Libertad:** Ofrece a los desarrolladores una gran libertad para usar y distribuir el software.

Desventajas:

1. **Falta de protección:** No protege contra el uso indebido del software.
2. **Sin copyleft:** No obliga a que las modificaciones también se distribuyan bajo la misma licencia.
3. **Ambigüedad en patentes:** No aborda explícitamente los derechos de patentes.
4. **Responsabilidad limitada:** No ofrece garantías ni asume responsabilidades por el software.
5. **Reutilización en software propietario:** Permite que el software se integre en proyectos propietarios sin compartir mejoras.

Referencia:

https://en.wikipedia.org/wiki/BSD_licenses

<https://snyk.io/learn/what-is-bsd-license/>

Curva de Aprendizaje: Baja a moderada. Su simplicidad y sintaxis clara facilitan el aprendizaje.

Ejemplo de Uso: Desarrollo de servidores web, microservicios y herramientas de línea de comandos.

Página Oficial: <https://go.dev/>