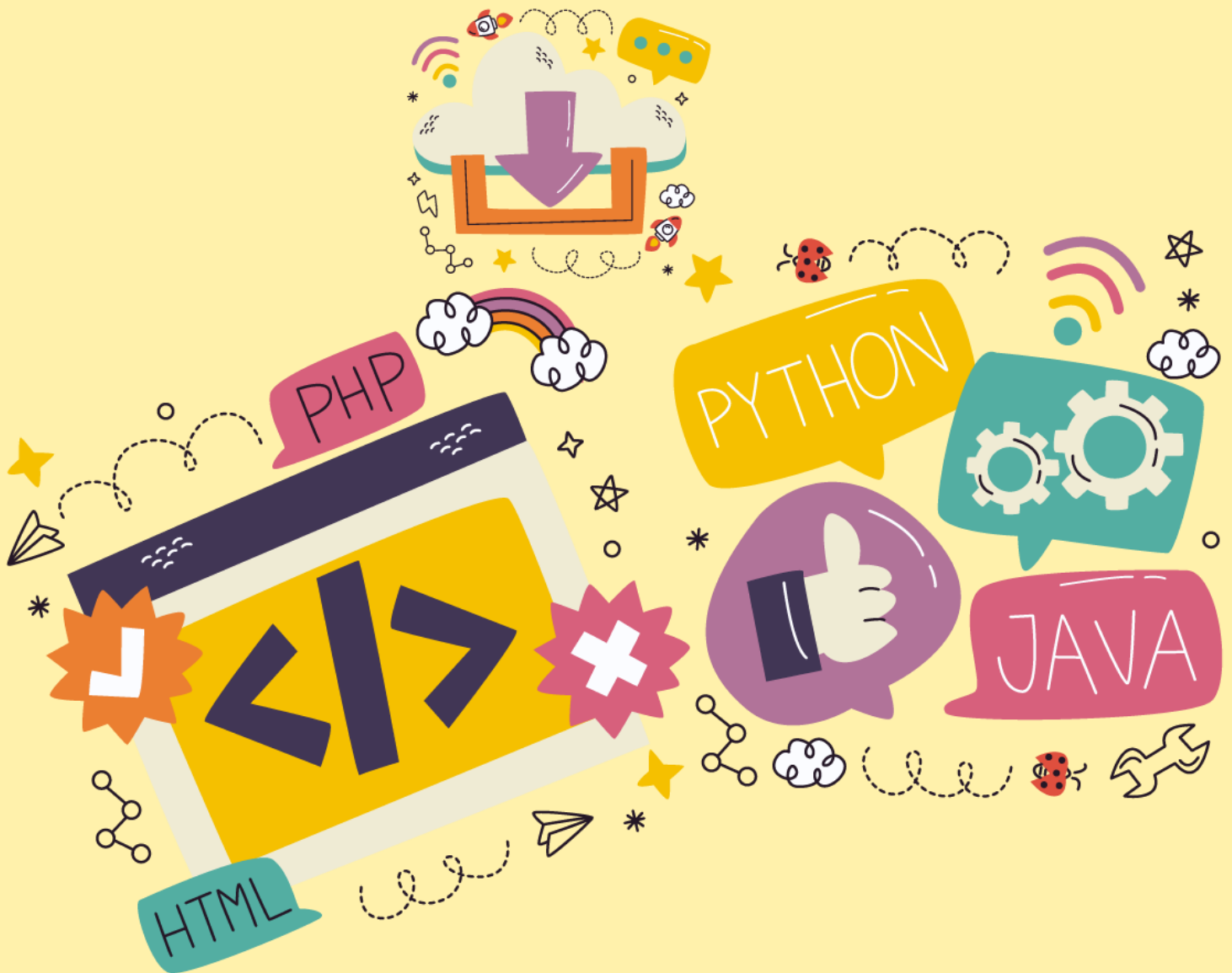


DESARROLLO WEB

(Lenguajes y frameworks)



INTEGRANTES:

Gonzalo Pozo

Ruben Peña

Hector Chango

Fang Yue Wu

Myriam Huamani

INDICE

PHP (HYPERTEXT PREPROCESSOR).....	3
1. HISTORIA Y EVOLUCIÓN	3
2. CARACTERÍSTICAS PRINCIPALES	3
3. FRAMEWORKS Y EXTENSIONES POPULARES	4
JAVASCRIPT.....	4
1. HISTORIA Y EVOLUCIÓN	4
2. EVOLUCIÓN:	5
3. CARACTERÍSTICAS PRINCIPALES	5
4. USO EN LA ACTUALIDAD	5
5. PLATAFORMAS EN LAS QUE SE PUEDE UTILIZAR	5
6. LENGUAJES DE PROGRAMACIÓN QUE SOPORTA.....	6
7. INTEGRACIÓN DE CONTROL DE VERSIONES	6
8. FRAMEWORKS POPULARES	6
9. MANUAL DE USUARIO Y DOCUMENTACIÓN.....	6
PYTHON.....	6
1. HISTORIA Y EVOLUCIÓN	6
2. PLATAFORMAS EN LAS QUE SE PUEDE UTILIZAR	7
3. LENGUAJES DE PROGRAMACIÓN QUE SOPORTA.....	7
4. INTEGRACIÓN DE CONTROL DE VERSIONES	7
5. RECONOCIMIENTO DE SINTAXIS Y EXTENSIONES	7
6. FRAMEWORKS POPULARES	7
7. MANUAL DE USUARIO Y DOCUMENTACIÓN.....	7
JAVA.....	8
1.HISTORIA Y EVOLUCIÓN	8
2. PLATAFORMAS DONDE SE PUEDE OPERAR	8
3. LENGUAJES DE PROGRAMACIÓN QUE SOPORTA.....	8
4. INTEGRACIÓN DE CONTROL DE VERSIONES	8
5. EXTENSIONES Y COMPONENTES.....	9
6. FRAMEWORKS MÁS POPULARES	9
6. MANUALES DE USUARIO Y AYUDA	9
7. VENTAJAS Y DESVENTAJAS DE USAR JAVA	9
RUBY.....	10
HISTORIA Y EVOLUCIÓN: VERSIONES EXISTENTES	10
PLATAFORMAS Y CONTROL DE VERSIONES	10
FRAMEWORKS QUE TIENEN Y CUÁLES SON LOS MÁS POPULARES.....	10
EXTENSIONES Y COMPONENTES PARA LA TECNOLOGÍA.....	11
MANUAL DE USUARIO Y AYUDA	11
VENTAJAS Y DESVENTAJAS QUE PRESENTAN FRENTE A OTROS COMPETIDORES	11
REFERENCIAS BIBLIOGRAFICAS.....	11

PHP (HYPERTEXT PREPROCESSOR)

1. HISTORIA Y EVOLUCIÓN

PHP es un lenguaje de programación interpretado del lado del servidor y de uso general que se adapta especialmente al desarrollo web. Fue creado inicialmente por el programador danés-canadiense Rasmus Lerdorf en 1994. En la actualidad, la implementación de referencia de PHP es producida por The PHP Group. PHP originalmente significaba Personal Home Page (Página personal), pero ahora significa el inicialismo recursivo PHP: Hypertext Preprocessor.

- PHP fue creado en 1994 por Rasmus Lerdorf. En sus primeras versiones, era solo un conjunto de scripts CGI (Common Gateway Interface) para gestionar páginas web sencillas. Desde entonces, ha evolucionado considerablemente: PHP 1.0 (1995): versión original creada por Rasmus Lerdorf para rastrear visitas a su página web.
- PHP 2.0 (1997): reescrito y conocido como PHP/FI (Personal Home Page/Forms Interpreter).
- PHP 3.0 (1998): introducción de nuevas capacidades orientadas a objetos. Fue la primera versión desarrollada por un equipo, con Zeev Suraski y Andi Gutmans.
- PHP 4.0 (2000): se basa en el motor Zend Engine y presenta mejoras significativas en rendimiento.
- PHP 5.0 (2004): introducción completa de la Programación Orientada a Objetos (POO), con soporte para interfaces y excepciones.
- PHP 6.0: fue un proyecto que estuvo en desarrollo entre 2005 y 2010, pero se retrasó y finalmente se canceló.
- PHP 7.0 (2015): gran mejora de rendimiento y consumo de memoria debido al motor Zend Engine 3.0. Se introducen nuevas características como los operadores de coalescencia y la declaración de tipo escalar.
- PHP 8.0 (2020): introduce JIT (Just-In-Time) para mejorar aún más el rendimiento, junto con mejoras de sintaxis y manejo de errores.

2. CARACTERÍSTICAS PRINCIPALES

En cuanto a plataformas, PHP es extremadamente versátil y puede ejecutarse en casi todas las plataformas principales, incluyendo: Windows, Linux, Unix y macOS. Además, a la hora de integrarse o desplegarse, PHP se puede usar en servidores web populares como Apache, Nginx e IIS.

- **PHP** se usa principalmente con su propio lenguaje de programación, pero soporta la interacción con otros lenguajes y tecnologías, como: HTML: para la estructura web.
- **JavaScript**: para interacción en el lado del cliente.
- **SQL**: para bases de datos.

Puede interactuar con Python, Java, C y C++ mediante extensiones o mediante ejecución de procesos externos.

PHP no incluye un sistema de control de versiones propio, pero se integra fácilmente con sistemas como Git. Los desarrolladores suelen usar estas herramientas para gestionar versiones de proyectos PHP, ya que permite rastrear cambios en el código fuente.

PHP soporta una amplia gama de extensiones y módulos, algunos de los más comunes son:

- **ocURL:** para hacer peticiones HTTP.
- **GD:** para manipulación de imágenes.
- **PDO:** para trabajar con bases de datos (MySQL, PostgreSQL, SQLite).
- **mbstring:** para manejo de cadenas multibyte.

3. FRAMEWORKS Y EXTENSIONES POPULARES

Las extensiones son parte fundamental del ecosistema PHP, y permiten ampliar sus funcionalidades.

Existen varios frameworks basados en PHP que ayudan a agilizar el desarrollo web y seguir buenas prácticas como la separación de lógica y presentación. Los más populares son:

- **Laravel:** el framework PHP más popular actualmente. Ofrece una estructura MVC (Model-View-Controller) robusta, migraciones de bases de datos, y funcionalidades como Eloquent ORM y Blade templating.
- **Symfony:** muy usado en aplicaciones empresariales. Enfocado en el rendimiento y la reutilización de componentes.
- **CodeIgniter:** ligero y fácil de aprender. Es ideal para aplicaciones pequeñas o medianas.

Existe una documentación de usuario oficial en la propia web de PHP (php.net/docs.php), en el apartado de Docs. Aunque, también podemos ayudarnos de sitios webs como W3Schools.

VENTAJAS:

- Fácil de aprender y usar: ideal para principiantes, especialmente si ya conocen HTML.
- Amplia comunidad y soporte: existe una vasta documentación, tutoriales y foros que facilitan el aprendizaje y resolución de problemas.
- Gran compatibilidad con servidores: PHP se integra fácilmente con servidores web como Apache y Nginx.

DESVENTAJAS:

- Historial de problemas de seguridad: aunque ha mejorado, las malas prácticas comunes pueden generar vulnerabilidades en las aplicaciones.
- Competencia con lenguajes más modernos: tecnologías como Node.js o Python (con Django/Flask) son vistas como más modernas y elegantes en algunos proyectos.
- Dificultad en proyectos grandes: manejar grandes aplicaciones con PHP puede volverse complejo en comparación con lenguajes más estrictos como Java o C#.

JAVASCRIPT

1. HISTORIA Y EVOLUCIÓN

Creación (1995): Brendan Eich, un desarrollador en Netscape creó JavaScript en solo 10 días bajo el nombre de Mocha. Luego se le cambió el nombre a LiveScript y finalmente a JavaScript para coincidir con la popularidad de Java en ese momento (aunque no están relacionados).

2. EVOLUCIÓN:

- **Estándar ECMAScript (1997):** ECMAScript es la especificación técnica subyacente de JavaScript, con la primera edición publicada en 1997 como ES1.
- **ES5 (2009):** Introdujo características importantes como el modo estricto, nuevos métodos de array, JSON y otras funcionalidades que lo hicieron más moderno y poderoso.
- **ES6 (2015):** Fue un hito en la evolución de JavaScript, agregando características clave como clases, promesas, deestructuración, módulos, y funciones flecha, lo que acercó a JavaScript a otros lenguajes de programación modernos.
- **Versión actual (2024):** JavaScript continúa evolucionando cada año con nuevas actualizaciones de ECMAScript, introduciendo mejoras como async/await y módulos dinámicos, manteniéndose a la vanguardia del desarrollo web.

3. CARACTERÍSTICAS PRINCIPALES

- **Lenguaje del lado del cliente:** JavaScript es un lenguaje interpretado que se ejecuta directamente en el navegador del usuario, permitiendo la creación de interfaces interactivas.
- **Flexibilidad:** Es un lenguaje multiparadigma, que admite programación imperativa, funcional y orientada a objetos.
- **Asincronía:** JavaScript ha sido optimizado para manejar operaciones asíncronas de forma eficiente, lo que lo hace ideal para aplicaciones modernas de tiempo real (e.g., chats, transmisiones en vivo).

4. USO EN LA ACTUALIDAD

- JavaScript es uno de los lenguajes más utilizados del mundo, especialmente en el desarrollo web, donde permite la creación de experiencias de usuario dinámicas y modernas. Algunos de sus usos más comunes incluyen:
- **Desarrollo frontend:** JavaScript es esencial para añadir interactividad a las páginas web. Los frameworks frontend como React, Vue.js, y Angular permiten construir interfaces complejas y escalables.
- **Desarrollo backend (Node.js):** Desde la aparición de Node.js, JavaScript se ha extendido al backend, permitiendo a los desarrolladores usar el mismo lenguaje tanto en el servidor como en el cliente.
- **Aplicaciones móviles y de escritorio:** Frameworks como React Native permiten desarrollar aplicaciones móviles con JavaScript, mientras que Electron es usado para crear aplicaciones de escritorio (e.g., Slack, VS Code).
- **Desarrollo de APIs y microservicios:** JavaScript con Node.js es utilizado para crear APIs rápidas y eficientes, ideales para sistemas de microservicios.

5. PLATAFORMAS EN LAS QUE SE PUEDE UTILIZAR

- **Navegadores Web:** Es su entorno principal, donde se usa para crear interactividad y dinamismo en las páginas web. Todos los navegadores modernos soportan JavaScript nativamente (Chrome, Firefox, Edge, Safari).

- **Servidores (Node.js):** Con Node.js, JavaScript se puede usar del lado del servidor para manejar solicitudes HTTP, acceder a bases de datos, y más. Es común en aplicaciones web de alto rendimiento y microservicios.
- **Aplicaciones Móviles (React Native):** JavaScript permite el desarrollo de aplicaciones móviles nativas para iOS y Android con herramientas como React Native.

6. LENGUAJES DE PROGRAMACIÓN QUE SOPORTA

- **TypeScript:** Un superconjunto de JavaScript que añade tipos estáticos. Facilita el desarrollo escalable y se convierte a JavaScript para ejecutarse en los navegadores.
- **WebAssembly (Wasm):** Un formato binario que permite ejecutar lenguajes como C, C++, Rust junto con JavaScript en el navegador.

7. INTEGRACIÓN DE CONTROL DE VERSIONES

Con plataformas como GitHub, GitLab o Bitbucket, los desarrolladores pueden colaborar en proyectos JavaScript, realizar pull requests, y gestionar el ciclo de vida del software.

8. FRAMEWORKS POPULARES

Frontend:

- **React.js:** Biblioteca para construir interfaces de usuario interactivas, creada por Facebook. Es muy usada por su enfoque en componentes y rendimiento.
- **Angular:** Framework robusto de Google para el desarrollo de aplicaciones de una sola página (SPA), usado en grandes aplicaciones empresariales.

Backend:

- **Node.js:** Plataforma que permite ejecutar JavaScript en el servidor, con una alta capacidad de manejar peticiones concurrentes.
- **Express.js:** Framework minimalista y flexible para Node.js, usado para crear APIs y aplicaciones web rápidas.

9. MANUAL DE USUARIO Y DOCUMENTACIÓN

- **MDN Web Docs:** La guía de referencia más completa y oficial sobre JavaScript, mantenida por Mozilla. Explica desde lo más básico hasta funcionalidades avanzadas, incluyendo ejemplos y tutoriales.

PYTHON

1. HISTORIA Y EVOLUCIÓN

- Python fue creado por Guido van Rossum y lanzado en 1991. El nombre proviene del grupo cómico Monty Python.
- **Python 1.0 (1994-2000):** Incluía soporte básico para módulos externos y comprensión de listas.
- **Python 2.0 (2000-2010):** Introdujo recolección de basura y comprensión de listas.

- **Python 3.0 (2008-Presente):** Incremento del `print()` para eliminar el uso de paréntesis en la sintaxis y la división entre enteros con float.
- La versión más reciente es **Python 3.11 (2022)**, que ofrece mejoras significativas de rendimiento.

2. PLATAFORMAS EN LAS QUE SE PUEDE UTILIZAR

- Python es apto para todas las plataformas. Podemos ejecutarlo en diferentes sistemas operativos como Windows, Linux y Mac, simplemente usando el intérprete correspondiente.

3. LENGUAJES DE PROGRAMACIÓN QUE SOPORTA

Aunque Python es un lenguaje en sí, se integra con otros lenguajes:

- **C/C++:** A través de CPython y Cython.
- **Java:** Con Jython para ejecutar Python en la JVM.

4. INTEGRACIÓN DE CONTROL DE VERSIONES

Python se integra perfectamente con sistemas de control de versiones como Git y Spyder es un IDE popular para Python que ofrece una amplia gama de herramientas y funcionalidades para facilitar el desarrollo de aplicaciones.

5. RECONOCIMIENTO DE SINTAXIS Y EXTENSIONES

- Python destaca por su sintaxis clara y uso de indentación en lugar de llaves para bloques de código.

EXTENSIONES:

- **PIP (installs packages):** Es una herramienta de línea de comandos que te permitirá instalar, reinstalar y desinstalar paquetes de Python, además de resolver automáticamente las dependencias de cada uno de estos paquetes.

6. FRAMEWORKS POPULARES

- **Django:** Framework completo para desarrollo web.
- **Flask:** Microframework para aplicaciones ligeras y flexibles.
- **FastAPI:** Para construir APIs rápidas y eficientes.

7. MANUAL DE USUARIO Y DOCUMENTACIÓN

- La documentación oficial está disponible en docs.python.org. Otros recursos útiles incluyen Stack Overflow y sitios como Real Python que ofrecen tutoriales.

VENTAJAS:

- **Facilidad de aprendizaje y legibilidad:** Su sintaxis simple y clara facilita el aprendizaje y hace que el código sea fácil de leer y mantener, ideal tanto para principiantes como para desarrolladores avanzados.

- Amplia comunidad y soporte de bibliotecas: Python tiene una gran comunidad activa que ofrece soporte, y cuenta con miles de bibliotecas disponibles para una variedad de aplicaciones, facilitando el desarrollo rápido.

DESVENTAJAS:

- Velocidad de ejecución: Python es más lento en comparación con lenguajes compilados como C o Java, ya que es interpretado y no compilado previamente.
- Menor eficiencia en gestión de memoria: Python no gestiona la memoria de manera tan eficiente como lenguajes de bajo nivel como C o C++, lo que puede ser una desventaja en aplicaciones críticas en rendimiento.

JAVA

1.HISTORIA Y EVOLUCIÓN

Java fue desarrollado por Sun Microsystems en 1995. Su objetivo inicial era simplificar la programación para dispositivos. Las versiones más importantes han sido las siguientes:

- **Java 1.0 (1996):** Primera versión oficial.
- **Java 2 (J2SE) (1998):** Introducción de la Máquina Virtual de Java (JVM) y tres ediciones (J2SE, J2EE y J2ME).
- **Java SE 5.0 (2004):** Nuevas características como genéricos (permite la creación de clases, interfaces y métodos) y autoboxing (permite la conversión automática entre tipos de datos primitivos (como int, float, double) y sus correspondientes clases envolventes (como Integer, Float, Double)).
- **Java 8 (2014):** Introducción de expresiones lambda (permiten escribir bloques de código que se pueden pasar como parámetros o usar directamente sin tener que crear una clase que implemente una interfaz).
- **Java 17 (2021):** Versión LTS (soporte a largo plazo), centrada en mejoras de seguridad y rendimiento.

2. PLATAFORMAS DONDE SE PUEDE OPERAR

Java es multiplataforma, tal cual permite su funcionamiento en casi todos los sistemas operativos y dispositivos tales como: Windows, Linux, macOS, Android (Android SDK), Dispositivos integrados (Java ME), Servidores web (Java EE).

3. LENGUAJES DE PROGRAMACIÓN QUE SOPORTA

Java es un lenguaje de programación por sí mismo, su JVM permite ejecutar otros lenguajes tales como: Kotlin, Groovy, Scala, Clojure, JRuby (implementación de Ruby en Java), Jython (implementación de Python en Java).

4. INTEGRACIÓN DE CONTROL DE VERSIONES

Java no incluye directamente un sistema de control de versiones, pero si soporta herramientas como:

- **Git:** Para gestión de código, gestión de versiones y ramas.

5. EXTENSIONES Y COMPONENTES

Java tiene extensiones y librerías para aumentar la capacidad del lenguaje. Estas son algunas de ellas:

- **Junit:** es la biblioteca o framework de pruebas más utilizada en Java.

Permite escribir pruebas unitarias para comprobar que el funcionamiento del código es el esperado.

- **Lombok:** Es una biblioteca muy conocida que reduce mucho el código repetido en Java. Ayuda a autogenerar código común como getters, setters, constructores, etc.
- **Maven/Gradle:** Se utiliza para la gestión de dependencias, como herramienta de construcción e incluso como herramienta de documentación. Es de código libre y gratuito.

6. FRAMEWORKS MÁS POPULARES

Java tiene numerosos frameworks que aceleran el desarrollo de aplicaciones. Los más conocidos incluyen:

- **Spring:** Framework para el desarrollo de aplicaciones web y empresariales, conocido por su flexibilidad y gestión de dependencias (Inyección de Dependencias).
- **Hibernate:** Utilizado para el mapeo objeto-relacional (ORM, es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia), facilitando la interacción con bases de datos.
- **JavaServer Faces (JSF):** Framework para desarrollar interfaces de usuario en aplicaciones web.
- **Apache Struts:** Framework de desarrollo web basado en el patrón MVC.

6. MANUALES DE USUARIO Y AYUDA

Existen numerosos recursos de documentación y ayuda oficial para Java estos son solo 2 ejemplos:

- **Oracle Documentation:** Documentación oficial de Java SE
- **Java Tutorials:** Una serie de tutoriales para principiantes y avanzados en la página oficial de Oracle.

7. VENTAJAS Y DESVENTAJAS DE USAR JAVA

Ventajas:

- **Multiplataforma:** Escribe una vez, ejecuta en cualquier lugar.
- **Gran comunidad:** Existen muchas bibliotecas y herramientas desarrolladas por la comunidad lo que le hace muy útil y fácil de trabajar ya que muchas cosas que queremos ya están creadas.
- **Seguridad:** La JVM proporciona un entorno controlado, protegiendo contra posibles ataques a la memoria.

- **Escalabilidad:** Es adecuado para aplicaciones de gran escala como sistemas bancarios y financieros. Ya que este funciona en casi todos los sistemas operativos.

Desventajas:

- **Rendimiento:** Aunque Java ha mejorado, su rendimiento puede ser inferior a lenguajes como C++ o Python debido a la ejecución en la JVM.
- **Curva de aprendizaje:** La sintaxis puede resultar más compleja en comparación con otros lenguajes más modernos y concisos como Python.

RUBY

HISTORIA Y EVOLUCIÓN: VERSIONES EXISTENTES

Este lenguaje fue creado por Yukihiro Matsumoto, quien empezó a trabajar en Ruby el 24 de febrero de 1993, y lo presentó al público en el año 1995.

Combina una sintaxis inspirada en Python y Perl, y combina características de programación orientada a objetos

PLATAFORMAS Y CONTROL DE VERSIONES

- **RubyMine:** IDE completo para Ruby y Rails de JetBrains, con excelente integración de Git para colaboración en equipo.
- **Visual Studio Code (VS Code):** Popular entre desarrolladores de Ruby por su flexibilidad y personalización. Ligero y con buena integración de Git, es ideal para Ruby on Rails.
- **Atom:** Editor de texto de código abierto, flexible y personalizable. Aunque menos funcional que otros IDE, se convierte en un buen entorno para Ruby on Rails con paquetes adicionales, ofreciendo resaltado de sintaxis y autocompletado.
- **Sublime Text:** Editor ligero y personalizable, valorado por su velocidad. Con paquetes específicos, se convierte en un IDE efectivo para Rails, ofreciendo funciones útiles como resaltado de sintaxis y búsqueda en el proyecto.

FRAMEWORKS QUE TIENEN Y CUÁLES SON LOS MÁS POPULARES.

Los principales frameworks de Ruby son:

- **Ruby on Rails (RoR):** El más popular para desarrollo web, sigue principios como "CoC" y "DRY" y usa el patrón MVC.
- **Sinatra:** Un microframework ligero y flexible, ideal para aplicaciones pequeñas y APIs simples.
- **Hanami:** Modular y ligero, con enfoque en rendimiento y simplicidad.
- **Padrino:** Basado en Sinatra, pero más completo, adecuado para aplicaciones web más grandes.
- **Grape:** Diseñado para construir APIs, fácilmente integrable con otros frameworks como Rails o Sinatra.

EXTENSIONES Y COMPONENTES PARA LA TECNOLOGÍA.

1. Visual Studio Code (VS Code)

- **Ruby:** Soporte básico para Ruby, incluyendo resaltado de sintaxis y fragmentos de código.
- **Ruby Solargraph:** Proporciona autocompletado, navegación de código e integración con el servidor de lenguaje Solargraph.
- **Ruby on Rails:** Ayuda con el desarrollo en Rails, incluyendo la navegación entre archivos y fragmentos útiles.

2. Atom

- **ide-ruby:** Proporciona autocompletado, linting y depuración para Ruby.
- **language-ruby:** Resaltado de sintaxis específico para Ruby.
- **ruby-test:** Para ejecutar pruebas de Ruby dentro de Atom.
- **linter-rubocop:** Linter para seguir las convenciones de código de Ruby usando Rubocop.

3. Sublime Text

Ruby on Rails Snippets: Fragmentos de código para acelerar el desarrollo en Rails.

SublimeLinter-rubocop: Integra Rubocop como linter.

RSpec: Mejora el soporte para pruebas con RSpec.

AutoRuby: Autocompletado para métodos Ruby.

MANUAL DE USUARIO Y AYUDA

En la sección de documentación de la página oficial de Ruby se encuentran los manuales de usuario de ayuda y tutoriales.

VENTAJAS Y DESVENTAJAS QUE PRESENTAN FRENTE A OTROS COMPETIDORES

Ruby es una excelente opción para el desarrollo web gracias a su sintaxis elegante, enfoque en la productividad, escalabilidad, comunidad activa y flexibilidad. Es ideal para construir aplicaciones web eficientes y de calidad.

Sin embargo, presenta desventajas en rendimiento, siendo más lento que lenguajes como Go o Java, y puede enfrentar problemas de escalabilidad en aplicaciones grandes, además de consumir más recursos y tener menor adopción fuera del desarrollo web, lo que dificulta encontrar desarrolladores expertos.

REFERENCIAS BIBLIOGRAFICAS

PHP

- <https://es.wikipedia.org/wiki/PHP>
- <https://en.wikipedia.org/wiki/PHP>
- <https://www.php.net/>
- <https://www.php.net/docs.php>
- <https://www.quora.com/Where-is-PHP-6>
- <https://www.arsys.es/blog/php>

PYTHON

- <https://www.luisllamas.es/historia-evolucion-python/>
- <https://docs.python.org/es/3/tutorial/index.html>
- <https://www.prodad-software.es/git-spyder/>
- <https://byspel.com/cual-framework-deberia-utilizar-en-python-flask-fastapi-o-django/>
- <https://keepcoding.io/blog/ventajas-y-desventajas-de-python/>

JAVASCRIPT

- <https://www.arsys.es/blog/extensiones-vscode>
- <https://keepcoding.io/blog/la-historia-de-javascript/>
- <https://blog.hubspot.es/website/ventajas-y-desventajas-de-javascript>

JAVA

- <https://www.campusmvp.es/recursos/post/java-que-es-maven-que-es-el-archivo-pom-xml.aspx>
- [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
- <https://junit.org/junit5/docs/current/user-guide/>
- <https://innowise.com/es/blog/benefits-and-drawbacks-of-java/>
- <https://blog.hubspot.es/website/ventajas-desventajas-java>

RUBY

- <https://platzi.com/blog/historia-ruby-on-rails/>
- <https://es.wikipedia.org/wiki/Ruby>
- <https://flatirons.com/blog/ruby-on-rails-ide/>
- <https://geekflare.com/es/top-ruby-frameworks/>
- <https://desarrolloweb.win/lenguaje-de-programacion-ruby-ventajas-y-desventajas/>