



MVC

Grupo #6

David Santiago Quintana – María José Niño R.

MVC

Modelo vista controlador

Ejemplo práctico

React
Spring boot
Redis

MVC

Historia

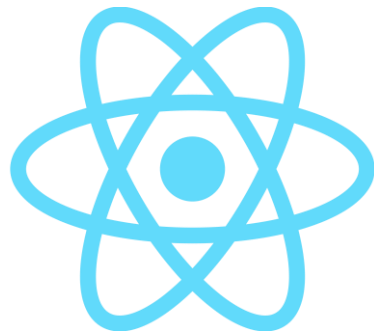
- 1970 introducido por Trygve Reenskaug
- Implementado para el lenguaje de programación Smaltalk-76

Ventajas

- Separa los datos
- Reutilización de código
- Facilita la realización de pruebas

Desventajas

- Aumento complejidad
- Aumento tiempo de desarrollo



Historia

- Desarrollada por la empresa Facebook.
- Lanzada en 2013, Jordan Walke

Ventajas

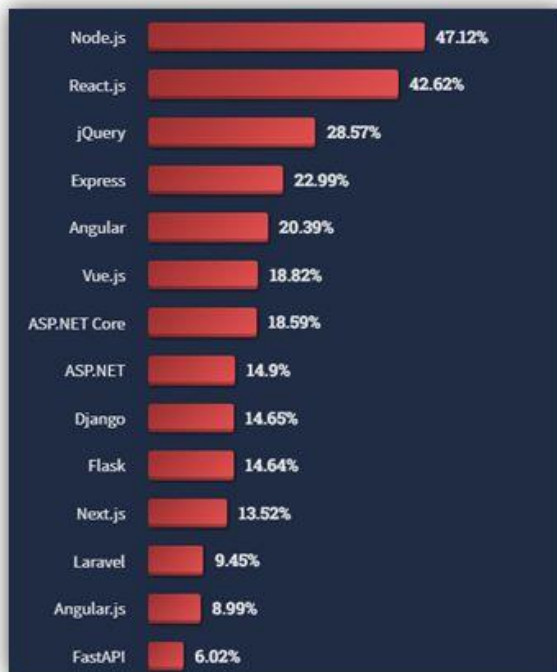
- Es fácil de aprender
- Alto nivel de flexibilidad y máxima capacidad de respuesta
- DOM virtual
- Vinculación de datos de manera descendente
- Biblioteca de código abierto

Desventajas

- Poca documentación oficial
- No hay un patrón de desarrollo

Actualización

Encuestas





Historia

- Desarrollado por el equipo de Spring Framework en 2013.
- Nace como forma para simplificar la configuración y uso de Spring Framework para los

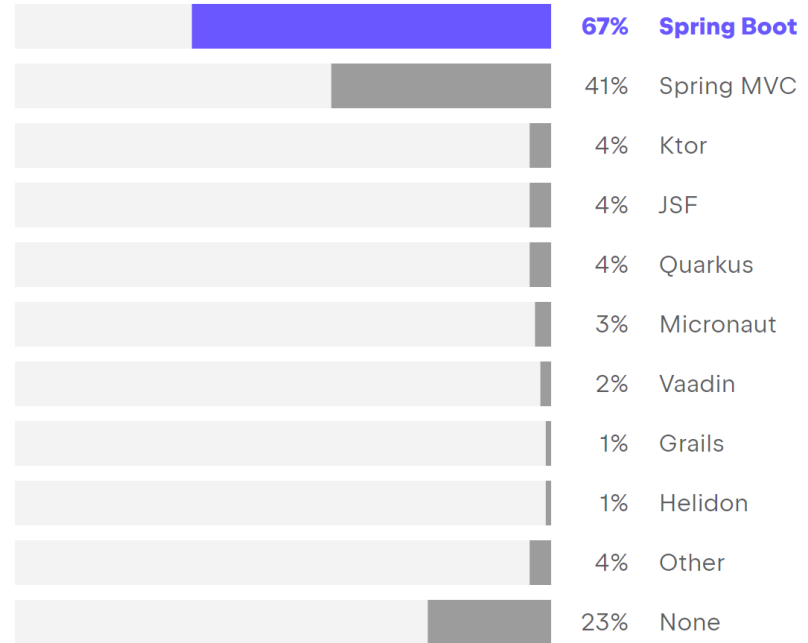
Ventajas

- configuración automática.
- Flexibilidad en el desarrollo de la aplicación
- Fácil integración con otras tecnologías
- Comunidad de desarrolladores

Desventajas

- Curva de aprendizaje
- Sobrecarga en el rendimiento

Encuestas





redis



Historia

- 2009 Salvatore Sanfilipp
- Empezó a trabajar en RDtool

Ventajas

- Los datos son almacenados en la RAM
- Admite una gran variedad de tipos de datos
- Permite agregar nodos adicionales
- Fácil de instalar y usar

Desventajas

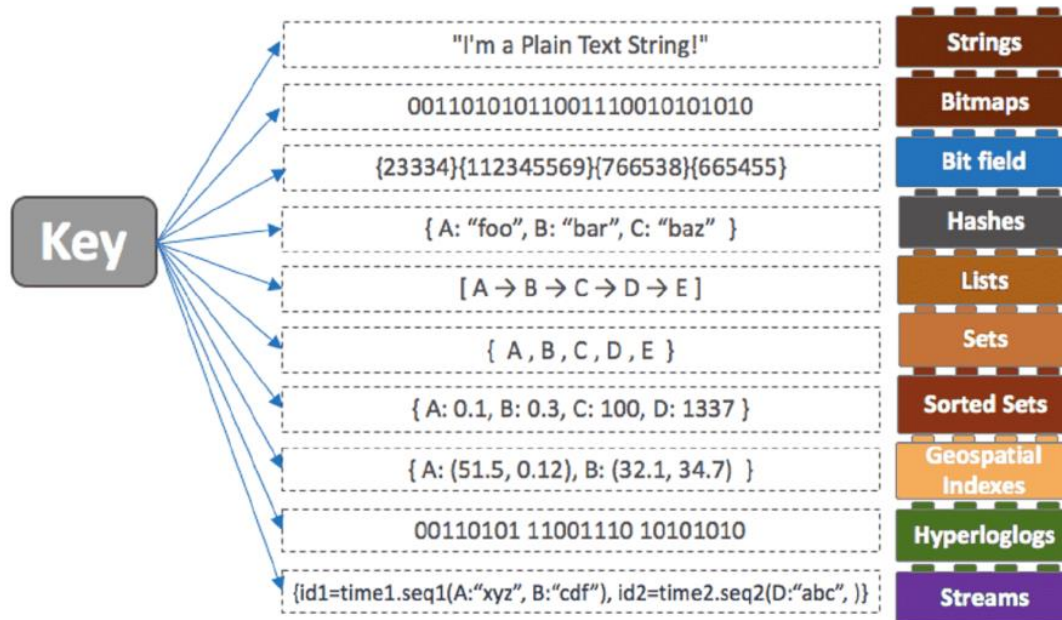
- Capacidad esta limitada por la memoria disponible en el sistema.
- Poca tolerancia a fallos

Amplia variedad de estructuras de datos

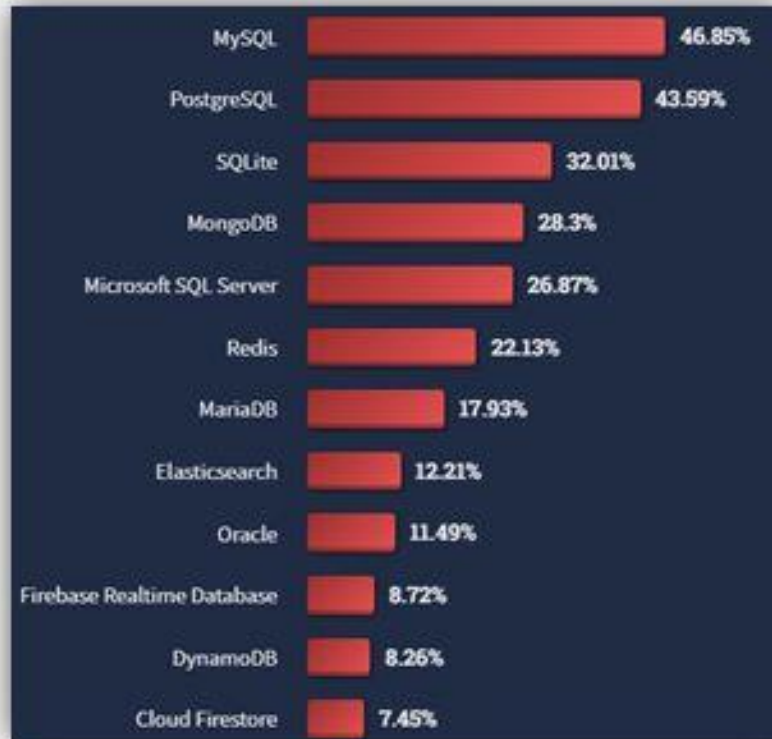
Hash, listas, sets, sorted sets, strings, bitmaps, entre otros.



Redis: Estructura de datos



Encuestas





Historia

- Ingenieros de Sun Microsystems, con James Gosling
- 1995 la empresa a lanzó Java como plataforma de desarrollo de software.

Ventajas

- Independiente de la plataforma
- Arquitectura de seguridad
- Amplia biblioteca
- Lenguaje orientado a objetos

Desventajas

- No tiene tan buen rendimiento
- La curva de aprendizaje

JS

Historia

- En los 90 la empresa Netscape desarrollaba su propio navegador y quería mejorar la interacción con el usuario
- En 1995 fue lanzada junto con el navegador de la misma empresa..

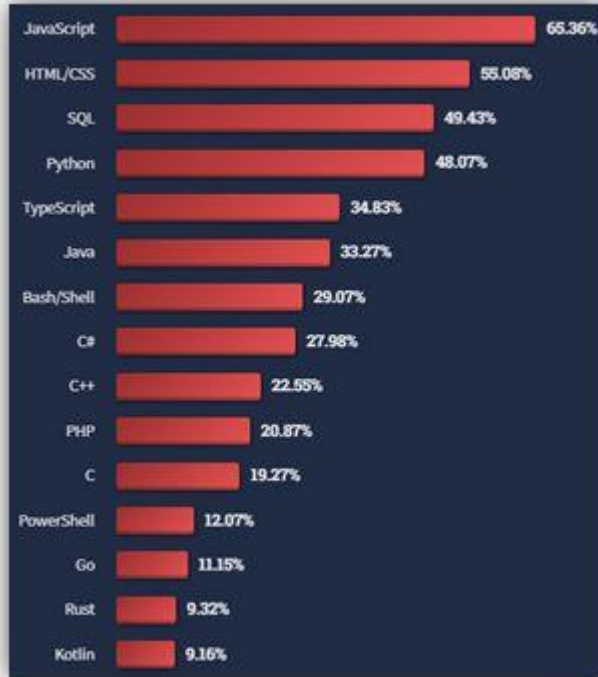
Ventajas

- Se puede utilizar en el lado del cliente y en el del servidor.
- Fácil de aprender
- Tiene una amplia biblioteca y diferentes frameworks
- Permite crear interfaces de usuario interactivas y dinámicas

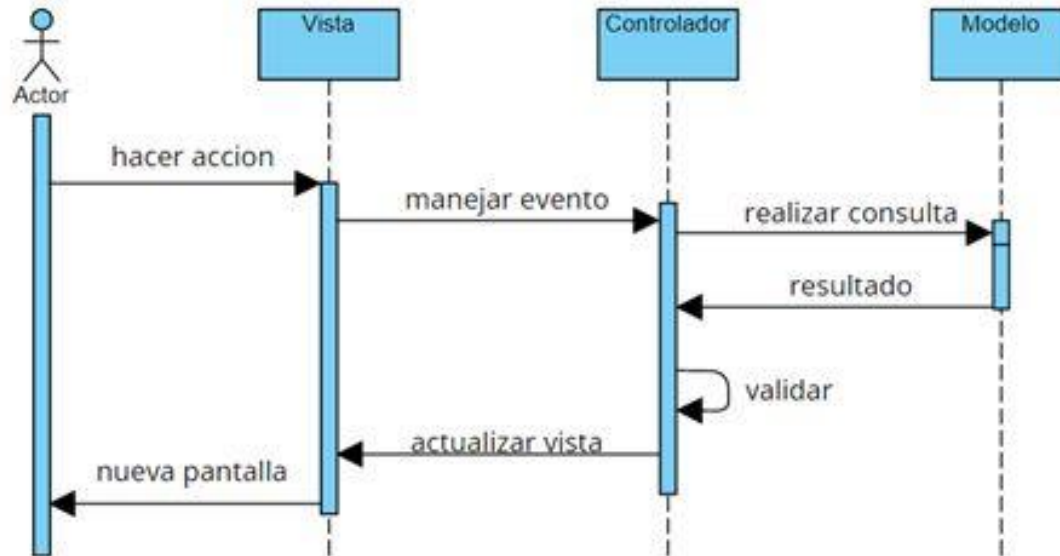
Desventajas

- implementación puede cambiar según el navegador
- Menos eficiente en cuanto a rendimiento.
- Puede ser vulnerable si no son implementadas las medidas de seguridad de manera correcta

Encuestas



UML



Principios SOLID

	SRP	OCP	LSP	ISP	DIP
Stack	Cada componente tiene una responsabilidad específica. Cada herramienta hace uso de este principio.	Permite la creación de nuevas funciones sin modificar el código existente. En React.js, se pueden utilizar patrones de composición para añadir nuevas funcionalidades a los componentes sin modificar su código original. En Spring Boot, se pueden utilizar patrones de inyección de dependencias para añadir nuevas funcionalidades a los servicios sin modificar su código original. En Redis, se pueden utilizar patrones de diseño de base de datos para extender la funcionalidad sin modificar los datos existentes.	Es posible reemplazar clases por sus subclasses sin afectar el comportamiento. En React.js, un componente que hereda de otro componente debe comportarse de manera similar al componente original y no cambiar su comportamiento básico. En Spring Boot, una subclase de una clase base debe poder ser utilizada en lugar de la clase base sin cambiar el comportamiento del sistema. En Redis, cualquier tipo de dato que se utilice para interactuar con la base de datos debe tener el mismo comportamiento que los tipos de datos existentes para evitar problemas de compatibilidad.		

Atributos de Calidad



	Escalabilidad	Mantenibilidad	Seguridad	Rendimiento	Modularidad	Confiabilidad	Usabilidad
Spring Boot	x	x	x				
React.js				x	x		x
Redis	x			x		x	
stack	x	x		x	x	x	



Patrones y estilos



Spring Boot	<ul style="list-style-type: none">- Patrón de arquitectura MVC, Spring Boot tiene soporte nativo para este patrón, facilitando la creación de aplicaciones web.- Estilo de arquitectura basado en microservicios, Spring Boot tiene soporte para la creación y gestión de microservicios, permitiendo la creación de aplicaciones escalables y fáciles de mantener.
React.js	<ul style="list-style-type: none">- Patrón de arquitectura Contenedor-Presentación, facilitando la reutilización de componentes y la gestión de la complejidad de aplicaciones grandes.- Estilo de arquitectura basado en componentes, React proporciona herramientas para la creación de componentes reutilizables y la gestión de su estado.
Redis	<ul style="list-style-type: none">- Patrón de arquitectura de caché, Redis tiene soporte nativo para la gestión de caché, facilitando la implementación.- Estilo de arquitectura basado en eventos, Redis tiene soporte para la gestión de eventos y la publicación y suscripción, permitiendo la creación de aplicaciones en tiempo real y sistemas de mensajería.

Aplicaciones

- ✓ Ruby
- ✓ Django
- ✓ Flask
- ✓ Adobe pothoshop
- ✓ Angry birds
- ✓ Instagram



MVC + STACK



Separación clara de
responsabilidades

Eficiencia en el rendimiento

Flexibilidad

Documentación y comunidad

Solución escalable, eficiente y flexible para el desarrollo
de aplicaciones web modernas y de alta demanda.



Ejemplo práctico



Diagrama alto nivel

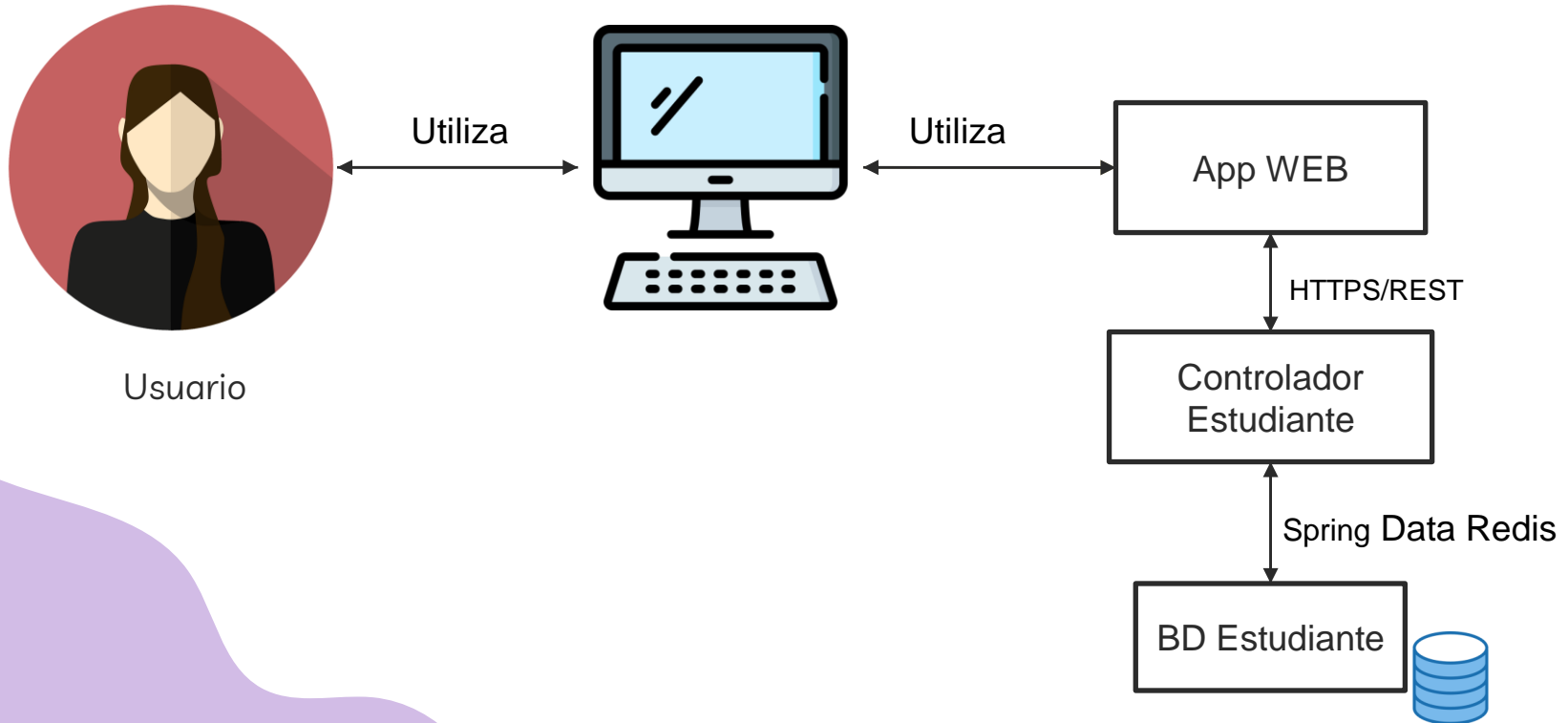
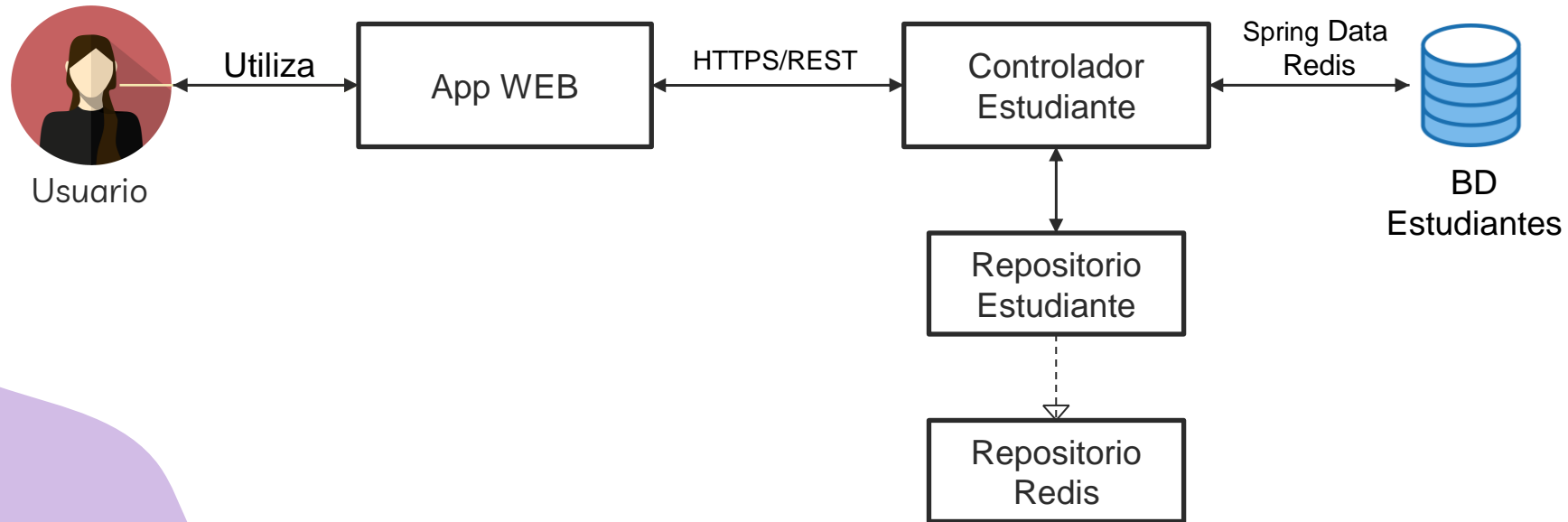


Diagrama bajo nivel





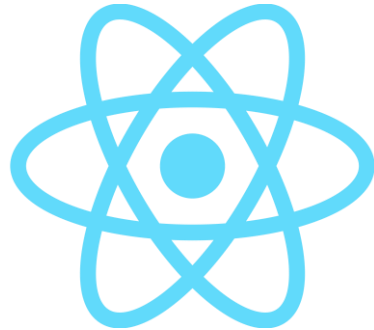
Gestor estudiantes

Crear

Nombre	Apellido	Correo electrónico	Estado	Acción
Santiago	Quintana	squintana@mail.co	Activado	Eliminar
Pepe	Ronaldo	si@no.es	Desactiva	Eliminar
Juan	Perez	hola@ok.com	Activo	Eliminar
Mario	Mendoza	m@hotmail.com	Activo	Eliminar
Maria	Barbosa	chao@nice.es	Desactiva	Eliminar

Conclusiones y lecciones aprendidas

MVC



redis



**MUCHAS
GRACIAS**

Referencias

- 1.React: A JavaScript library for building user interfaces. (n.d.). Retrieved March 23, 2023, from <https://reactjs.org/>
- 2."React (JavaScript library)", Wikipedia, [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library)), consultado en marzo 23, 2023.
- 3."A Brief History Of React", Smashing Magazine, <https://www.smashingmagazine.com/2020/10/brief-history-of-react/>, consultado en marzo 23, 2023.
- 4."Spring Boot", Spring.io, <https://spring.io/projects/spring-boot>, consultado en marzo 23, 2023.
- 5."The History of Spring Boot", DZone, <https://dzone.com/articles/the-history-of-spring-boot>, consultado en marzo 23, 2023.
- 6."Spring Boot Overview", Baeldung, <https://www.baeldung.com/spring-boot-overview>, consultado en marzo 23, 2023.
- 7.Redis, Redis Labs, <https://redis.io/>, consultado en marzo 23, 2023.
- 8."Redis", Wikipedia, <https://en.wikipedia.org/wiki/Redis>, consultado en marzo 23, 2023.
- 9."Redis: A Brief History and Overview", DigitalOcean, <https://www.digitalocean.com/community/tutorials/redis-a-brief-history-and-overview>, consultado en marzo 23, 2023.
- 10."Java Overview", Oracle, <https://www.oracle.com/java/overview/>, consultado en marzo 23, 2023.
- 11."A Brief History of Java", DZone, <https://dzone.com/articles/a-brief-history-of-java>, consultado en marzo 23, 2023.
- 12."Java Programming Language", GeeksforGeeks, <https://www.geeksforgeeks.org/java/>, consultado en marzo 23, 2023.
- 13."JavaScript", Mozilla Developer Network, <https://developer.mozilla.org/en-US/docs/Web/JavaScript>, consultado en marzo 23, 2023.
- 14."JavaScript", W3Schools, <https://www.w3schools.com/js/>, consultado en marzo 23, 2023.
- 15.Keepcoding (2021). La historia de JavaScript. Recuperado el 24 de marzo de 2023, de <https://keepcoding.io/blog/la-historia-de-javascript/#:~:text=El%20origen%20javascript%20surge%20a,lo%20que%20hac%C3%ADa%20el%20usuario>.
- 16.JetBrains (2022). Estado del ecosistema de desarrollo de 2022. Recuperado el 24 de marzo de 2023, de <https://www.jetbrains.com/es-es/lp/devecosystem-2022/>
- 17.Stack Overflow (2022). Encuesta de Stack Overflow 2022. Recuperado el 24 de marzo de 2023, de <https://survey.stackoverflow.co/2022/#most-popular-technologies-database>
- 18.State of JS (2022). State of JS 2022: Usage. Recuperado el 24 de marzo de 2023, de <https://2022.stateofjs.com/en-US/usage/>
- 19.JetBrains (2022). Estado del ecosistema de desarrollo de 2022. Recuperado el 24 de marzo de 2023, de <https://www.jetbrains.com/es-es/lp/devecosystem-2022/>
- 20.Stack Overflow (2022). Encuesta de Stack Overflow 2022. Recuperado el 24 de marzo de 2023, de <https://survey.stackoverflow.co/2022/#most-popular-technologies-database>
- 21.State of JS (2022). State of JS 2022: Usage. Recuperado el 24 de marzo de 2023, de <https://2022.stateofjs.com/en-US/usage/>
- 22.Martin, R. C. (2003). The Principles of OOD. Recuperado el 24 de marzo de 2023, de <https://www.butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>
- 23.Scotch Development (2019). S.O.L.I.D: Los primeros cinco principios del diseño orientado a objetos. Recuperado el 24 de marzo de 2023, de <https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design>