

**FACULTAD DE INGENIERÍA**

**DEPARTAMENTO DE INGENIERÍA DE SISTEMAS**

**ARQUITECTURA DE SOFTWARE**

**RESULTADOS DE INVESTIGACIÓN PATRONES**

**TALLER 2: PATRÓN ARQUITECTURAL MVC**

**JUAN ALEJANDRO DIAZ LOTE**

**ANDRES FELIPE DORADO**

**MARIANA ROJAS GALAVÍS**

**02 DE ABRIL 2023**

Contenido

[1. Definición, historia y evolución de los temas asignados (Usar diagramas y/o infografías que expliquen los temas, si aplica para el tema). 3](#_Toc131330602)

[2. Relación entre los temas asignados. 6](#_Toc131330603)

[3. Situaciones y/o problemas donde se pueden aplicar los temas. 7](#_Toc131330604)

[4. Para Estilos y patrones: representación UML de estructura, comportamiento e interacción (ej. diagrama de clases, máquinas de estado y secuencia). 9](#_Toc131330605)

[5. Ventajas y Desventajas de cada uno de los temas. 10](#_Toc131330606)

[6. Principios SOLID. 13](#_Toc131330607)

[7. Atributos de Calidad. 14](#_Toc131330608)

[8. Casos de estudio relevantes (casos de aplicación) 15](#_Toc131330609)

[9. Ejemplo práctico y funcional relacionando los temas 16](#_Toc131330610)

[10. Conclusiones 17](#_Toc131330611)

[11. Lecciones aprendidas 17](#_Toc131330612)

[12. Referencias 17](#_Toc131330613)

**Sistema:** Monolítico MVC

**Front-end:** Angular (JavaScript)

**Back-end:** Spring-boot (Java)

**Base de datos:** MariaDB

## **Definición, historia y evolución de los temas asignados (Usar diagramas y/o infografías que expliquen los temas, si aplica para el tema).**

* 1. **MVC**

El patrón arquitectónico MVC se centra en la separación de la lógica de negocio y la interfaz de usuario para permitir la evolución independiente de ambos aspectos, lo que mejora la mantenibilidad con el modularidad que implica su uso y la fiabilidad con la separación de responsabilidades.

El patrón MVC consta de tres capas o componentes principales. En primer lugar, están los modelos, que representan los objetos de negocio y se encargan de todo lo relacionado con los datos, desde la extracción hasta la actualización. Esta capa es el núcleo funcional de la aplicación. En segundo lugar, están las vistas, que representan la interfaz de usuario o los sistemas que interactúan con la aplicación. Esta capa organiza y selecciona los datos para mostrarlos al usuario de manera visual. La capa de vistas está estrechamente relacionada con la capa de modelos, ya que solicita los datos de la capa de modelos para mostrarlos al usuario. Finalmente, están los controladores, que gestionan el flujo de trabajo de la aplicación. Esta capa se encarga de establecer la relación entre las capas de modelos y vistas, y define cuándo y cómo se solicitan y muestran los datos. Los controladores implementan las acciones que se pueden realizar en la aplicación, como editar, buscar o interactuar, y actúan como intermediarios entre las capas de modelos y vistas.

Diagrama

Descripción generada automáticamente

Ilustración 1. Funcionamiento MVC

Este patrón fue creado en la década de 1970 por Trygve Reenskaug, un informático noruego, mientras trabajaba como investigador en el Laboratorio de Investigación de Xerox de Palo Alto . La idea principal era proporcionar una arquitectura que permitiera a los usuarios interactuar con una aplicación desde diferentes perspectivas sin afectar el funcionamiento.

Una década después, la arquitectura MVC se popularizó con la llegada de las GUI (interfaces gráficas de usuario) en los sistemas operativos de escritorio como el Macintosh y el Windows. Esta arquitectura se convirtió en una forma estándar de estructurar aplicaciones de software con una interfaz gráfica. Para 1990 y tras el surgimiento de la web y la popularidad de las aplicaciones web, se dio la evolución de MVC para adaptarse a esta nueva plataforma. La capa de vistas se volvió más importante, ya que era necesario entregar contenido HTML dinámico en lugar de una interfaz gráfica estática.

La popularidad de los frameworks de desarrollo web, como Ruby on Rails y Django, llevó a una mayor estandarización en la implementación de MVC en aplicaciones web para los 2000. Ruby on Rails es una aplicación que ha sido desarrollada desde 2003 y proporciona una estructura web de código abierto basada en Ruby para la creación de aplicaciones prácticas. Utiliza el patrón de arquitectura MVC para organizar la programación de aplicaciones y todas sus capas están diseñadas para integrarse perfectamente. A través de este framework, se puede programar todo, desde plantillas para controlar el flujo de la aplicación hasta la lógica de negocios, en Ruby.

Además para esta década también empezaron a surgir diferentes variantes del modelo como **MVVM y MVP.**

A propósito de las diferentes variantes del modelo y con el auge de las aplicaciones móviles, **MVC** ha evolucionado desde 2010 para satisfacer las nuevas necesidades. Se han desarrollado diferentes framework como React que utiliza una variante de MVC que se llama **Flux**.

* 1. **Angular:**

Angular es un framework de Javascript especialmente adecuado para crear aplicaciones frontend modernas, que sean de complejidad media o elevada. Con Angular, se pueden desarrollar aplicaciones del tipo SPA (Single Page Application), así como las conocidas como PWA (Progressive Web App). A continuación, se describen las versiones desde su creación:

* AngularJS (2010): La primera versión de Angular, conocida como AngularJS, fue lanzada en 2010 y se centró en el desarrollo de aplicaciones de una sola página (SPA) con características como vinculación de datos, inyección de dependencias y directivas personalizadas.
* Angular 2 (2016): Angular 2 introdujo una arquitectura completamente nueva y una reescritura completa del código base de AngularJS. Introdujo una nueva sintaxis, un enfoque en los componentes y la introducción de Angular CLI.
* Angular 4 (2017): Angular 4 se centró en mejorar el rendimiento y la facilidad de uso. Introdujo el uso de operadores de RxJS y mejoras en la animación.
* Angular 5 (2017): Angular 5 se centró en mejorar el rendimiento y la facilidad de uso. Introdujo mejoras en el enrutamiento y una nueva forma de procesar la carga diferida.
* Angular 6 (2018): Angular 6 introdujo la característica de carga diferida, que permite cargar módulos solo cuando se necesitan. También se mejoró el soporte para Progressive Web Apps (PWA) y el rendimiento.
* Angular 7 (2018): Angular 7 se centró en mejoras en el rendimiento y la facilidad de uso. Introdujo mejoras en el enrutamiento y la capacidad de aplicar directivas personalizadas a elementos nativos.
* Angular 8 (2019): Angular 8 se centró en mejorar la velocidad y el rendimiento de la aplicación. Introdujo mejoras en la compilación y el enrutamiento.
* Angular 9 (2020): Angular 9 se centró en mejorar la experiencia del desarrollador y la velocidad de compilación. Se introdujo Ivy Renderer y mejoras en la modularidad de la aplicación.
* Angular 10 (2020): Angular 10 se centró en mejorar el rendimiento y la facilidad de uso. Introdujo mejoras en Ivy Renderer y mejoras en el enrutamiento.
  1. **Spring Boot:**

Esta herramienta facilita la creación de aplicaciones web y microservicios utilizando el framework Spring, permitiendo que el proceso de desarrollo sea más sencillo y rápido. Respecto a su historia, a continuación, se presenta un breve recuento del hito que dio a su origen:

En 2012, se presentó un problema en el repositorio de GitHub de Spring con el título "Mejora del soporte para arquitecturas de aplicaciones web sin contenedor". El autor del problema, Mike Youngstrom, propuso simplificar la arquitectura de las aplicaciones web de Spring, eliminando la necesidad de un contenedor de servlets y utilizando en su lugar el modelo de componentes y configuración de Spring desde arriba hacia abajo.

Phil Webb, miembro del equipo de Spring, decidió crear un nuevo proyecto llamado Spring Boot que abordara este problema y otros. Spring Boot hace que sea fácil crear aplicaciones y servicios con Spring con el mínimo esfuerzo y ha sido ampliamente adoptado.

Versiones:

* Spring Boot surgió, como se mencionó anteriormente, cuando en octubre de 2012, un cliente llamado Mike Youngstrom hizo una solicitud en Jira pidiendo que se iniciara rápidamente el framework de Spring. Y así, a principios de 2013, se creó Spring Boot.
* En abril de 2014, se creó Spring Boot 1.0, seguido de varias versiones: Spring Boot 1.1 en junio de 2014, 1.2 en marzo de 2015, 1.3 en diciembre de 2016, 1.4 en enero de 2017 y Spring Boot 1.5 en febrero de 2017.
  1. **MariaDB**

MariaDB es un sistema de gestión de bases de datos relaciona es decir, un conjunto de programas que permiten modificar, almacenar, y extraer información de una base de datos, desarrollado por Michael Widenius (fundador de MySQL), cuyo prinicipal diferencial es que es de uso completamente libre.

La historia de MariaDB se remonta a 2009, cuando se anunció la adquisición de MySQL AB por parte de Sun Microsystems, que a su vez fue adquirida por Oracle Corporation en 2010. Algunos miembros del equipo original de desarrollo de MySQL, liderados por Michael "Monty" Widenius, se preocuparon por la dirección que estaba tomando MySQL bajo la propiedad de Oracle y decidieron crear una bifurcación para continuar el desarrollo de una versión independiente de MySQL, que eventualmente se convirtió en MariaDB.

Desde su creación, MariaDB se ha mantenido como una base de datos de código abierto y ha evolucionado para incluir nuevas características y funcionalidades que no están disponibles en MySQL, como la capacidad de trabajar con bases de datos distribuidas y la compatibilidad con diferentes motores de almacenamiento. Además, se ha enfocado en la seguridad, la fiabilidad y el rendimiento, y ha ganado una amplia aceptación en la comunidad de desarrolladores de bases de datos.

En la actualidad, MariaDB es utilizado por muchas empresas, organizaciones y comunidades en todo el mundo, y cuenta con una amplia gama de herramientas y aplicaciones que lo hacen fácil de usar y administrar. Además, cuenta con un equipo de desarrollo activo y una comunidad de usuarios comprometida que sigue trabajando para mejorar y expandir las capacidades de la base de datos.

Hasta la versión 5.5 de MariaDB, esta base de datos seguía el esquema de numeración de versiones de MySQL, buscando así garantizar la compatibilidad con la misma versión principal de MySQL. Sin embargo, a partir del año 2012, se empezaron a añadir cada vez más funcionalidades a MariaDB que no estaban disponibles en MySQL. Esto llevó a que se cambiara el sistema de numeración de versiones de MariaDB Server y se lanzara la versión 10.0, mientras que MySQL lanzó la 5.6. Actualmente, la versión de soporte a largo plazo (LTS) más reciente es MariaDB 10.6, mientras que la última versión estable de soporte a corto plazo es MariaDB 10.9.

* 1. **Java**

Java es un lenguaje de programación orientado a objetos creado en la década de 1990 por James Gosling y su equipo en Sun Microsystems. Originalmente, Java fue diseñado para ser una alternativa a los lenguajes de programación C y C++, y fue lanzado al mercado en 1995.

Java se destacó desde su lanzamiento como un lenguaje único debido a que su propia traducción compilaba e interpretaba de forma paralela, lo que lo diferenciaba de otras opciones de programación. Esta diferencia generó un gran impacto y convirtió a Java en una de las opciones más populares para programación. En un principio, se utilizó principalmente para conectar sistemas en oficinas y otros espacios de comunicación, pero al cambiar su enfoque hacia el mundo online, Java entró de lleno en la web.

En poco tiempo, otras opciones de programación comenzaron a ganar popularidad en el mercado, lo que llevó a Java a perder terreno gradualmente en el mundo online. En el año 2016, Oracle, la empresa que adquirió a Sun Microsystems en 2010, decidió disminuir el papel de Java y retiró muchos de sus subprogramas en poco tiempo. No obstante Java es un lenguaje de programación de gran relevancia en todo el mundo, con una comunidad amplia y diversa que se extiende por todos sus componentes y con más de cuatro millones de desarrolladores en todo el mundo. Debido a su popularidad, millones de dispositivos utilizan Java en la actualidad.

* 1. **JavaScript**

JavaScript es un lenguaje de programación que se utiliza en el desarrollo de páginas web para crear experiencias interactivas para los usuarios. Con la ayuda de JavaScript, se pueden agregar diversas funciones a un sitio web, como actualizar las fuentes de redes sociales, mostrar animaciones y crear mapas interactivos.

JavaScript fue creado en 1995 por Brendan Eich en Netscape, con el objetivo inicial de ser un lenguaje de scripting para el navegador Netscape Navigator. En aquel entonces, la web estaba en pañales y las páginas eran estáticas, sin embargo, Netscape se propuso mejorar la experiencia del usuario con la inclusión de un lenguaje de scripting que permitiera interactuar con la página de forma dinámica. El nombre inicial de JavaScript era Mocha, pero luego fue cambiado a LiveScript y finalmente a JavaScript por motivos comerciales.

JavaScript se popularizó rápidamente debido a su capacidad de permitir a los desarrolladores crear aplicaciones web interactivas por lo que empezaron a su vez a surgir diferentes versiones por lo que ECMA International decidió que todos los navegadores debían usar un estándar de JavaScript.Desde entonces es la responsable de añadir nuevas características de tal forma que cada año una nueva versión de JavaScript bajo el nombre de ECMAScript.

* 1. **SQL**

El SQL es un lenguaje de programación que se utiliza para gestionar y manipular información almacenada en una base de datos relacional. En una base de datos de este tipo, la información se almacena en forma de tabla, en la que cada fila y columna representa diferentes atributos y relaciones de datos. A través de las instrucciones SQL, es posible realizar diversas operaciones como el almacenamiento, actualización, eliminación, búsqueda y recuperación de información.

En 1969, el investigador de IBM, Edgar F. Codd, desarrolló el modelo de base de datos relacional que consistía en la asociación de "claves" con diferentes datos, como un nombre de usuario con un número de teléfono y un nombre real. IBM creó un lenguaje para los sistemas de gestión de bases de datos relacionales basado en el trabajo de Codd, llamado SEQUEL, que más tarde se convertiría en SQL después de varias revisiones y mejoras.

Las pruebas iniciales de SQL se realizaron en 1978 y en 1981 IBM lanzó su primer producto comercial, SQL/DS, seguido de DB2 en 1983. Otros proveedores como Sybase, Ingres y Oracle, que lanzó su primer producto en 1979, también adoptaron SQL como lenguaje de programación para la gestión de bases de datos relacionales.

## **Relación entre los temas asignados.**

Java es un lenguaje de programación orientado a objetos, muy popular y utilizado en una amplia variedad de aplicaciones empresariales, desde aplicaciones de escritorio hasta servidores web y aplicaciones móviles. Es conocido por su seguridad, portabilidad y robustez, lo que lo convierte en una opción popular para empresas y desarrolladores de software.

JavaScript, por otro lado, es un lenguaje de programación interpretado que se ejecuta en el navegador web del cliente y se utiliza principalmente para crear aplicaciones web interactivas y dinámicas. Con el tiempo, se ha convertido en uno de los lenguajes de programación más populares y versátiles, con un amplio conjunto de bibliotecas y frameworks que simplifican su uso y aumentan su capacidad.

Angular es un framework de JavaScript desarrollado y mantenido por Google, que se utiliza para construir aplicaciones web de una sola página (SPA) y aplicaciones móviles híbridas. Proporciona una estructura completa para el desarrollo de aplicaciones, incluyendo características como enrutamiento, inyección de dependencias, validación de formularios, pruebas y más. Angular se basa en el patrón MVC para separar la lógica de negocio, la presentación y el control de la aplicación, lo que facilita su mantenimiento y escalabilidad.

Spring Boot, por otro lado, es un framework de Java que se utiliza para desarrollar aplicaciones web y de servidor. Proporciona una infraestructura completa para el desarrollo de aplicaciones Java, incluyendo características como la gestión de dependencias, la configuración automática, el servidor integrado y la seguridad. Spring Boot se basa en el patrón MVC para separar la lógica de negocio, la presentación y el control de la aplicación, lo que lo hace adecuado para construir aplicaciones web escalables y robustas.

MariaDB es una base de datos relacional de código abierto que se utiliza para almacenar y gestionar datos. Es una bifurcación de MySQL y es compatible con la mayoría de los sistemas de gestión de bases de datos relacionales. MariaDB es una alternativa popular y de alta calidad a otras bases de datos relacionales, y se utiliza a menudo en combinación con frameworks como Spring Boot y Angular.

El patrón MVC es un patrón de diseño arquitectónico utilizado en el desarrollo de aplicaciones de software. Se basa en la separación de la lógica de negocio, la presentación y el control de la aplicación en tres componentes principales: Modelo, Vista y Controlador. El Modelo se encarga de la lógica de negocio y la gestión de datos, la Vista se encarga de la presentación de la información al usuario y el Controlador se encarga de gestionar la interacción del usuario con la aplicación.

En resumen, Java y JavaScript son lenguajes de programación utilizados en diferentes contextos, y Angular y Spring Boot son frameworks que se basan en el patrón MVC para construir aplicaciones web escalables y robustas. MariaDB es una base de datos relacional de código abierto que se utiliza comúnmente con frameworks como Angular y Spring Boot. El patrón MVC es un patrón de diseño arquitectónico clave utilizado en el desarrollo de aplicaciones de software modernas. En conjunto, estos elementos forman una poderosa combinación de tecnologías y herramientas que permiten a los desarrolladores construir aplicaciones empresariales escalables, robustas y de alta calidad. Si bien cada una de estas tecnologías puede ser utilizada por separado, cuando se combinan adecuadamente y se utilizan de manera efectiva, pueden ayudar a los desarrolladores a construir aplicaciones complejas con mayor eficiencia y eficacia.

## **Situaciones y/o problemas donde se pueden aplicar los temas.**

* 1. **MVC**

El patrón de arquitectura Modelo-Vista-Controlador (MVC) es un patrón muy popular y ampliamente utilizado en el desarrollo de software, especialmente en aplicaciones web. La principal ventaja de este patrón es la clara separación de responsabilidades que permite, lo que facilita la organización y el mantenimiento de la aplicación a medida que aumenta su complejidad y tamaño. El patrón MVC se puede aplicar en diversas situaciones, incluyendo el desarrollo de aplicaciones web, móviles y de escritorio, así como en el desarrollo de juegos y en la integración de sistemas. En el caso de aplicaciones web, el patrón permite separar la lógica del negocio, la presentación de datos y la interacción del usuario en tres componentes diferentes. El modelo representa la lógica del negocio y la capa de acceso a datos, la vista es responsable de la presentación de datos al usuario, y el controlador maneja la interacción del usuario con la aplicación. En el desarrollo de aplicaciones móviles, el modelo representa la lógica del negocio y la capa de acceso a datos, la vista muestra los datos al usuario y el controlador maneja los eventos del usuario. En aplicaciones de escritorio, el patrón se puede utilizar para separar la lógica del negocio y la presentación de datos de la interfaz de usuario y los controladores de eventos. En el desarrollo de juegos, el patrón MVC también se puede aplicar para separar la lógica del juego, la presentación gráfica y la interacción del usuario. Finalmente, en la integración de sistemas, el patrón MVC puede ser útil cuando se necesitan capas separadas para la lógica del negocio y la presentación de datos. En resumen, el patrón MVC es ampliamente utilizado en el desarrollo de software en situaciones donde se necesita una separación clara de responsabilidades y una organización estructurada de la aplicación.

* 1. **Angular**

Angular es un framework de desarrollo de aplicaciones web que se puede utilizar en una amplia variedad de situaciones y problemas. Es especialmente útil para crear aplicaciones de una sola página (SPA), aplicaciones empresariales, aplicaciones móviles híbridas, aplicaciones de comercio electrónico, dashboards y paneles de administración, aplicaciones de juegos y aplicaciones de redes sociales. En el caso de las SPA, Angular utiliza el enrutamiento del lado del cliente para cambiar el contenido de la página sin tener que cargar una página nueva. Para las aplicaciones empresariales, ofrece una gran cantidad de características útiles, como la gestión de datos y la integración con otras herramientas y servicios. En las aplicaciones móviles híbridas, Angular se utiliza junto con herramientas como Ionic Framework para crear aplicaciones que se ejecutan en dispositivos móviles. Para las aplicaciones de comercio electrónico, Angular permite la creación de interfaces de usuario interactivas y fáciles de usar, y la integración con herramientas de pago y carrito de compras. En cuanto a los dashboards y paneles de administración, Angular es una opción popular debido a su capacidad para crear interfaces de usuario personalizadas y la integración con herramientas de visualización de datos. En el caso de las aplicaciones de juegos, Angular proporciona una estructura útil para la organización del código de juego y para la gestión de la interfaz de usuario. Por último, en las aplicaciones de redes sociales, Angular se utiliza para crear interfaces de usuario interactivas y la integración con herramientas de autenticación y de compartir en redes sociales. En resumen, Angular es una herramienta versátil y útil para desarrollar una amplia variedad de aplicaciones web.

* 1. **Spring Boot**

Spring Boot es un marco de trabajo en Java que se utiliza ampliamente en la creación de aplicaciones empresariales, microservicios y sistemas de procesamiento de datos. En particular, Spring Boot es una opción popular para el desarrollo de aplicaciones web empresariales, ya que ofrece una amplia gama de características para el desarrollo web, como la creación de controladores, servicios y capas de persistencia de datos, seguridad, gestión de transacciones, integración con bases de datos y muchas más. Además, Spring Boot es una excelente opción para crear microservicios, que son pequeños servicios independientes que funcionan juntos para cumplir una tarea específica. Spring Boot ofrece muchas herramientas y características para facilitar la creación, la ejecución y el monitoreo de microservicios. Otras situaciones y problemas donde se puede aplicar Spring Boot incluyen el procesamiento de datos, especialmente en el ámbito empresarial, la creación de sistemas de análisis de datos en tiempo real, sistemas de informes y dashboards, y la creación de aplicaciones móviles escalables. En particular, Spring Boot se puede utilizar para crear una API RESTful que se comunica con una aplicación móvil o para crear una aplicación móvil híbrida utilizando herramientas como Ionic Framework. En resumen, Spring Boot es una herramienta versátil y poderosa para el desarrollo de una amplia variedad de aplicaciones empresariales, microservicios y sistemas de procesamiento y análisis de datos. Ofrece una amplia gama de características y herramientas para facilitar el desarrollo de aplicaciones de alta calidad y escalables en Java.

* 1. **MariaDB**

MariaDB es una base de datos relacional de código abierto que se ha convertido en una herramienta muy popular para el almacenamiento y gestión de datos. La popularidad de esta herramienta radica en su capacidad para ser utilizada en diversas situaciones y problemas que requieran el manejo de grandes cantidades de información.

Por ejemplo, en el ámbito empresarial, MariaDB se puede utilizar para almacenar y gestionar grandes cantidades de datos relacionales, lo que facilita la realización de análisis y consultas complejas. También es una excelente opción para integrarse con otras aplicaciones empresariales.

En el desarrollo de aplicaciones web y móviles, MariaDB se ha convertido en una herramienta de elección para los desarrolladores. Muchos marcos de trabajo modernos, como Ruby on Rails, Django, Laravel y Spring Boot, son compatibles con MariaDB, lo que significa que los desarrolladores pueden utilizar esta herramienta para construir aplicaciones escalables y robustas.

En el ámbito de los sitios de comercio electrónico, MariaDB se utiliza comúnmente para almacenar y gestionar grandes cantidades de datos relacionados con los productos, pedidos y clientes. La capacidad de MariaDB para manejar grandes cantidades de datos relacionales, lo hace ideal para esta tarea.

Finalmente, en el análisis de datos, MariaDB se puede utilizar para realizar análisis complejos en grandes conjuntos de datos. Las empresas pueden utilizar MariaDB para analizar grandes cantidades de datos y obtener información valiosa sobre sus operaciones y clientes.

* 1. **Java**

Java es un lenguaje de programación de alto nivel muy utilizado en el desarrollo de aplicaciones, y puede ser utilizado en una variedad de situaciones y problemas. En el desarrollo de aplicaciones de escritorio, Java es una buena opción para crear aplicaciones que se ejecutan en el sistema operativo del usuario. Esto puede ser útil en situaciones donde se requiere una interfaz gráfica de usuario compleja, como en aplicaciones de software de contabilidad o de gestión de proyectos. En el desarrollo de aplicaciones web, Java es ampliamente utilizado para crear aplicaciones web dinámicas y escalables. Los desarrolladores pueden utilizar Java para crear aplicaciones web que sean útiles para una variedad de fines, como aplicaciones de comercio electrónico, aplicaciones de gestión de inventario, entre otros. Algunos de los marcos de trabajo más populares para el desarrollo web en Java son Spring, Hibernate y Struts. En el desarrollo de aplicaciones móviles, Java también es una opción popular. A través del uso de Android Studio, los desarrolladores pueden crear aplicaciones móviles para la plataforma Android utilizando Java. Esto permite a los desarrolladores crear aplicaciones móviles utilizando la misma plataforma de desarrollo que utilizan para aplicaciones de escritorio y web. En el desarrollo de juegos, Java es un lenguaje de programación popular para juegos en línea y para dispositivos móviles. Algunos de los juegos más populares, como Minecraft y Runescape, han sido desarrollados utilizando Java.

Finalmente, en el análisis de datos, Java también puede ser utilizado. Los desarrolladores pueden crear programas de análisis de datos complejos utilizando Java, que puede manejar grandes conjuntos de datos.

* 1. **JavaScript**

JavaScript es un lenguaje de programación que se utiliza principalmente en el desarrollo de aplicaciones web y para agregar interactividad a las páginas web. Es un lenguaje muy versátil que se utiliza en una variedad de situaciones y problemas.

Una de las aplicaciones más comunes de JavaScript es la validación de formularios en páginas web, lo que ayuda a reducir errores y mejorar la calidad de la información que se recopila. Además, JavaScript se utiliza ampliamente en el desarrollo de aplicaciones web interactivas, lo que permite a los desarrolladores crear aplicaciones web que ofrezcan una experiencia de usuario rica y dinámica.

Otra aplicación de JavaScript es la programación de animaciones en páginas web. Esto puede incluir animaciones de desplazamiento, transiciones de página y efectos de imagen. Además, JavaScript se utiliza para desarrollar juegos en línea, y algunos juegos populares como Angry Birds y Candy Crush Saga han sido creados con JavaScript.

Finalmente, JavaScript también se puede utilizar para automatizar tareas repetitivas en las páginas web, como el llenado automático de formularios, la selección de opciones y la navegación en sitios web.

* 1. **SQL**

El lenguaje SQL es un lenguaje de consulta estructurado que se utiliza para manejar bases de datos relacionales. En términos generales, se aplica en situaciones donde se requiere administrar grandes cantidades de datos y organizarlos de manera eficiente.

SQL se utiliza en la creación de bases de datos y en la definición de la estructura de las tablas, lo que permite el almacenamiento y la organización de datos de manera efectiva. Además, también se utiliza para realizar consultas a bases de datos, lo que incluye la búsqueda de datos específicos y la realización de operaciones de filtrado y ordenación para obtener información relevante.

Asimismo, SQL se aplica para modificar y actualizar los datos almacenados en una base de datos. Es posible realizar tareas como la eliminación de datos no deseados y la inserción de nuevos datos, entre otros. Además, también se utiliza para analizar los datos almacenados en una base de datos, lo que incluye la realización de cálculos y estadísticas para obtener información relevante.

Otra aplicación importante de SQL es la gestión de usuarios y permisos en una base de datos. Esto permite controlar quién puede acceder a la información almacenada en una base de datos y quién puede modificarla.

## **Para Estilos y patrones: representación UML de estructura, comportamiento e interacción (ej. diagrama de clases, máquinas de estado y secuencia).**

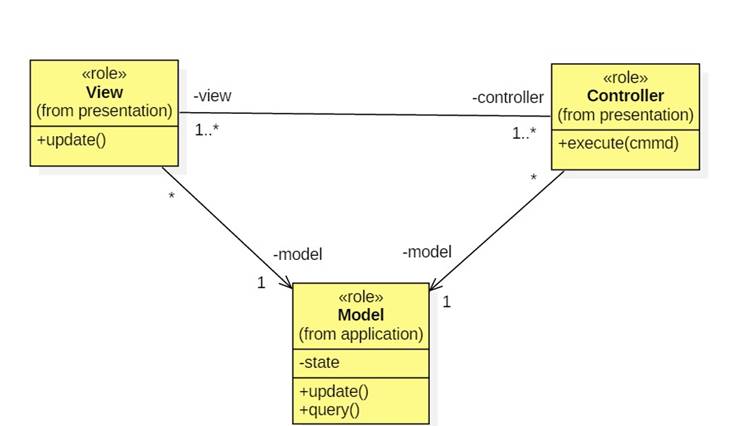


Ilustración 2. Diagrama de clases Patrón MVC

En el diagrama, hay tres componentes principales: Modelo, Vista y Controlador. El Modelo representa los datos y la lógica subyacentes de la aplicación, y es responsable de administrar los datos e implementar las reglas de negocio. La Vista es responsable de presentar los datos al usuario en un formato gráfico. El Controlador actúa como intermediario entre el Modelo y la Vista, maneja la entrada del usuario y actualiza el Modelo y la Vista según sea necesario.

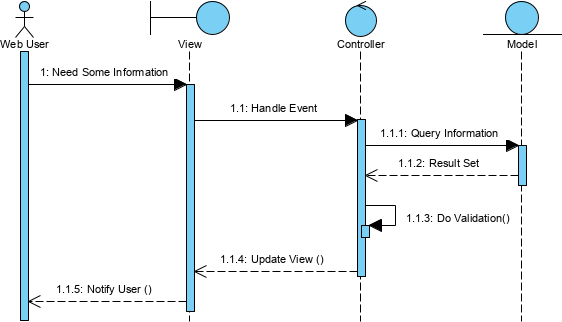


Ilustración 3. Ejemplo diagrama de secuencia MVC

Lo que se puede observar en el diagrama anterior es que un usuario web inicia una consulta y genera un evento que es procesado por el controlador para obtener la información necesaria del modelo, validarla y pasar el conjunto de resultados de regreso a la vista.

Para el caso del proyecto el diseño que se siguió es el siguiente:

Por la parte del back-end:

Diagrama

Descripción generada automáticamente

Ilustración 4. Diagrama clases

El diagrama anterior evidencia la relación entre las clases en la parte del back-end desarrollado con Spring Boot. Del diagrama podemos identificar que cuando se envía una solicitud HTPP, el controlador (sea el del equipo o el de los conductores conforme sea la solicitud), para realizar cualquier tipo de acción sea crear, eliminar, modificar u obtener, se comunica con el servicio de la respectiva entidad pasándole esta misma, este a su vez es el encargado de comunicarse con la parte de persistencia ,que se añadió implementando el patrón de Repository para de esta manera a su vez bajar el acoplamiento.

En adición en esta parte de implementó la inyección de dependencias al utilizar una capa para el servicio y una para el repositorio (las cuales se incluyeron para continuar con la separación de responsabilidades). Tenemos entonces que la capa del controlador depende de la de servicio para realizar la lógica de negocio y la capa de servicio depende de la capa de repositorio para realizar la persistencia de los datos.

Diagrama

Descripción generada automáticamente

Ilustración 5. Diagrama Secuencia ejemplo

En la ilustración 5 se puede ver el ejemplo de la secuencia de cómo sería la interacción cuando un usuario quiere solicitar la información sobre un conductor en específico. En primer lugar la vista le transmitiría la solicitud del usuario al controlador quien se comunicaría a su vez con el servicio y este con el repositorio para obtener la información. Luego y una vez la solicitud realizada sea la correcta, se devuelve el modelo del conductor solicitado hasta el controlador quien refresca la vista.

## **Ventajas y Desventajas de cada uno de los temas.**

* 1. **MVC**

Beneficios de MVC:

1. La separación de responsabilidades permite crear independencia del funcionamiento por lo que la implementación se realiza de forma modular.

2. Es un patrón que se puede adaptar a diferentes frameworks.

3. Es más fácil realizar pruebas unitarias y de integración.

Desventajas MVC:

1. A medida que crece el programa, la cantidad de archivos a mantener y desarrollar aumenta considerablemente.

2. La curva de aprendizaje del patrón puede ser más alta que otros modelos por lo que los programadores deben aprender cómo implementarlo adecuadamente para su funcionamiento eficiente.

3.Es un patrón de diseño orientado a objetos por lo que su implementación es difícil en lenguajes que no sigan este paradigma.

* 1. **Angular**

Beneficios de usar Angular:

1. Modularidad: Angular le permite estructurar el código de su aplicación en términos de módulos, lo que facilita su mantenimiento y ampliación.

2. Rendimiento: Con Angular, puede aumentar su productividad utilizando el amplio conjunto de herramientas y bibliotecas disponibles en el marco. Además, cuenta con una comunidad sólida y activa que comparte conocimientos y recursos para ayudarlo a crecer más rápido.

3. Inyección de dependencia: Angular utiliza una inyección de dependencia potente que facilita la organización y el uso compartido del código de la aplicación.

4. Compatibilidad: Angular es compatible con otros marcos y bibliotecas, lo que facilita la integración de otras funciones y herramientas en su aplicación.

Desventajas de usar Angular:

1. Curva de aprendizaje: Angular tiene una curva de aprendizaje relativamente pronunciada en comparación con otros marcos de JavaScript. Puede ser difícil para los principiantes aprender todo lo que necesitan para usar Angular de manera efectiva.

2. Comandos complejos. Algunos comandos y herramientas de Angular pueden ser confusos y difíciles de usar, especialmente si es nuevo en el sistema.

3. Tamaño: Angular es un marco grande y pesado en comparación con otros marcos livianos, lo que afecta el rendimiento de la aplicación.

4. Compatibilidad con versiones anteriores: las actualizaciones angulares hacen que la aplicación sea incompatible con versiones anteriores del marco, lo que puede llevar mucho tiempo y esfuerzo para actualizar la aplicación.

* 1. **Spring Boot**

Beneficios de usar Spring Boot:

1. Eficiente: Spring Boot tiene una configuración predeterminada que le permite iniciar rápidamente el desarrollo de aplicaciones sin configurar muchas cosas. Además, cuenta con muchas librerías y herramientas para aumentar tu productividad.

2. Modularidad: Spring Boot le permite estructurar su código en módulos, lo que facilita el mantenimiento y la extensibilidad.

3. Inyección de dependencia: Spring Boot utiliza una potente inyección de dependencia para organizar y compartir fácilmente el código de su aplicación.

4. Compatibilidad: Spring Boot es compatible con otras tecnologías y plataformas de Java, lo que facilita la integración de otras funciones y herramientas en su aplicación.

Desventajas de usar Spring Boot:

1.Curva de aprendizaje: Spring Boot tiene una curva de aprendizaje relativamente pronunciada en comparación con otras plataformas Java. Puede ser difícil para los principiantes aprender todo lo necesario para usar Spring Boot de manera efectiva.

2. Ajuste la configuración. Aunque Spring Boot tiene una configuración predeterminada, a veces es necesario ajustar la configuración de acuerdo con los requisitos específicos de la aplicación, lo que puede requerir conocimientos adicionales.

3. Sobrecarga: Spring Boot puede aumentar la carga de una aplicación debido a la gran cantidad de funciones y bibliotecas que contiene.

4. Compatibilidad con versiones anteriores: las actualizaciones de Spring Boot pueden hacer que su aplicación sea incompatible con versiones anteriores del marco, lo que puede llevar mucho tiempo y esfuerzo para actualizar su aplicación.

* 1. **MariaDB**

Beneficios de usar MariaDB:

1. Licencia de código abierto: MariaDB es una base de datos de código abierto, lo que significa que puede usarla y modificarla de forma gratuita.

2. Compatibilidad con MySQL: MariaDB es compatible con MySQL, lo que significa que puede migrar fácilmente de MySQL a MariaDB sin cambios importantes en el código.

3. Escalabilidad. MariaDB puede manejar grandes cantidades de datos y es escalable, lo que significa que puede agregar más servidores para manejar más tráfico.

4. Alto rendimiento: MariaDB proporciona un rendimiento rápido y eficiente, ideal para aplicaciones que requieren acceso y procesamiento de datos rápidos.

5. Seguridad: MariaDB ofrece una gran seguridad y puede cifrar sus datos para mantener su información privada y protegida.

Desventajas de usar MariaDB:

1. Complejidad. Configurar y mantener MariaDB puede ser complicado, especialmente si es nuevo en las bases de datos de código abierto.

2. Restricciones de almacenamiento. MariaDB tiene un límite en la cantidad de datos que puede almacenar, lo que puede limitar el desarrollo de aplicaciones.

3. No compatible. Como base de datos de código abierto, es posible que MariaDB no tenga el mismo nivel de soporte que las bases de datos comerciales y, cuando lo tiene, puede ser difícil obtener ayuda. Algunas aplicaciones pueden no ser compatibles con MariaDB, por lo que es posible que sea necesario ajustar el código de la aplicación.

## **Principios SOLID.**

Angular:

* Principio de responsabilidad única (SRP): los componentes angulares tienen una responsabilidad única y una razón única para cambiar. Cada componente es responsable de administrar la lógica de esa parte de la aplicación, como presentar datos o interactuar con el usuario.
* Principio abierto/cerrado (OCP): en Angular, los módulos y servicios están diseñados para estar abiertos a la extensión pero cerrados a la modificación. Esto significa que se pueden agregar nuevas funciones sin cambiar el código existente.
* Principio de sustitución de Liskov (LSP): Angular sigue el principio de sustitución de Liskov, que le permite heredar e implementar interfaces en componentes y servicios. Esto significa que puede crear componentes y servicios para reemplazar otros componentes y servicios sin cambiar el comportamiento de su aplicación.
* Principio de inversión de dependencia (DIP): Angular utiliza la inyección de dependencia para hacer que los módulos y servicios de alto nivel dependan de abstracciones en lugar de dependencias concretas. Esto permite reemplazar componentes y servicios sin afectar el resto de la aplicación.

Spring Boot:

* Principio de responsabilidad única (SRP): en Spring Boot, los componentes y servicios tienen una responsabilidad y una razón para cambiar. Cada componente es responsable de manejar la lógica de parte de la aplicación, como presentar datos o interactuar con los usuarios.
* Principio abierto/cerrado (OCP): en Spring Boot, los módulos y servicios están diseñados para estar abiertos a extensiones pero cerrados a modificaciones. Esto significa que se pueden agregar nuevas características sin cambiar el código existente.
* Principio de sustitución de Liskov (LSP): Spring Boot sigue el principio de sustitución de Liskov, que permite la herencia y la implementación de interfaces entre beans y servicios. Esto significa que puede crear componentes y servicios que pueden ser reemplazados por otros componentes y servicios sin cambiar el comportamiento de su aplicación.
* Principio de inversión de dependencia (DIP): Spring Boot utiliza la inyección de dependencia para que los módulos y servicios de nivel superior puedan basarse en abstracciones en lugar de dependencias concretas. Esto permite reemplazar componentes y servicios sin afectar el resto de la aplicación.

MariaDB:

* Principio de responsabilidad única (SRP): en MariaDB, SRP se puede aplicar al diseño de tablas y campos de bases de datos para garantizar que cada tabla y campo tenga una responsabilidad única y dedicada para almacenar un tipo específico de datos.
* Principio de apertura/cierre (OCP): la política OCP de MariaDB se puede utilizar para crear vistas y procedimientos almacenados que le permiten ampliar la funcionalidad de su base de datos sin modificar directamente las tablas y campos existentes.
* Principio de sustitución de Liskov (LSP): el LSP de MariaDB puede crear relaciones entre tablas utilizando índices y claves externas. Esto le permite agregar nuevas tablas y relaciones sin cambiar el comportamiento de las consultas existentes.
* Principio de inversión de dependencia (DIP): las políticas DIP de MariaDB se pueden aplicar a vistas y procedimientos almacenados que encapsulan la lógica de acceso a la base de datos. Esto garantiza que las consultas de la aplicación dependan de abstracciones en lugar de consultas SQL específicas, lo que facilita el uso compartido de consultas y el mantenimiento de la aplicación.

## **Atributos de Calidad.**

Los atributos de calidad que presenta el que se pueden encontrar en el stack y con el patrón son:

* Usabilidad: al separar la lógica de presentación de la lógica de negocio, MVC permite la creación de interfaces de usuario más intuitivas y amigables. Además Angular es un framework de JavaScript que ofrece una amplia gama de características para mejorar la usabilidad de las aplicaciones web del lado del cliente.
* Mantenibilidad: el patrón MVC permite la fácil modificación, actualización y prueba de componentes individuales sin afectar a la totalidad de la aplicación, lo que facilita la mantenibilidad del sistema lo anterior gracias a la separación de preocupaciones que implica el patrón.
* Eficiencia de desempeño: la estructura de MVC puede mejorar la eficiencia de la aplicación, ya que los componentes de la aplicación están diseñados para cumplir con una tarea específica.
* Adecuación Funcional: el patrón MVC permite separar las responsabilidades de la aplicación en tres componentes distintos, lo que facilita la implementación de funcionalidades específicas de manera clara y organizada.
* Fiabilidad: Spring Boot proporciona una amplia gama de herramientas y funciones para garantizar la fiabilidad de las aplicaciones.
* Portabilidad: el patrón se pude adaptar a diferentes software y en adición Spring Boot, Angular y MariaDB se puede utilizar en diferentes plataformas y entornos
* Spring Boot proporciona características de seguridad como la autenticación y autorización, la gestión de tokens y la configuración de permisos de acceso a los recursos de la aplicación

## **Casos de estudio relevantes (casos de aplicación)**

* 1. **Ruby On Rails:** es un framework de desarrollo web de código abierto que utiliza el lenguaje de programación Ruby. Se basa en el patrón de arquitectura MVC (Modelo-Vista-Controlador), que ayuda a los desarrolladores a organizar su código de manera efectiva y separar las preocupaciones en sus aplicaciones web.

Entre las ventajas de este framework encontramos:

* La comunidad de Ruby on Rails es muy activa y apasionada, lo que significa que siempre habrá un lugar al cual acudir en caso de necesitar ayuda o tener alguna duda.
* Se puede convertir las ideas en un producto mínimo viable de manera rápida y sencilla, ya que su tiempo de desarrollo es muy bajo.
* Al contar con una gran comunidad, es muy probable que alguien ya haya solucionado cualquier problema técnico que se presente.
* La sintaxis de Ruby on Rails es sencilla y fácil de leer, lo que facilita el trabajo de los desarrolladores.

Algunas de las empresas que usan Ruby On Rails son: airbnb, Github y Twitch.

* 1. **Catalyst:** Es un framework open source con una estructura diseñada para aplicaciones web, la cual destaca por su sencillez y flexibilidad. Los desarrolladores pueden usar Catalyst para construir aplicaciones web o para conectarse con módulos Perl que realicen tareas específicas necesarias para el correcto funcionamiento de la aplicación. Catalyst se inspira en otros frameworks similares, como Ruby on Rails o Spring, y sigue el patrón de diseño basado en MVC.

Entre las ventajas de este framework encontramos:

* Es versátil que ofrece una amplia gama de modelos de base de datos y es compatible con varios lenguajes de plantilla.
* Incluye su propio servidor de pruebas ligero para el desarrollo, que se reinicia automáticamente cuando se realizan cambios en el código fuente.
* Es compatible con todos los principales servidores web, gracias a su soporte nativo para PSGI/Plack.
* Cuenta con una gran cantidad de plugins listos para usar para la gestión de sesiones, autenticación de usuarios, caché y mucho más, lo que evita tener que crear todo desde cero.

Algunas de las empresas que usan Catalyst son:BBC iplayer, FreeCycle, Manchester Evening News.

* 1. **CakePHP:** es un framework de desarrollo de aplicaciones web de código abierto en el lenguaje de programación PHP. El framework proporciona una estructura para desarrollar aplicaciones web de manera rápida y sencilla utilizando el patrón de diseño Modelo-Vista-Controlador (MVC).

Entre sus ventajas encontramos:

* Permite construir prototipos rápidamente con características de generación de código y andamiaje, sin la necesidad de archivos XML o YAML complicados.
* Sus convenciones de Modelo-Vista-Controlador (MVC) limpias le permiten desarrollar su aplicación de manera más rápida y fácil, sin tener que planificar dónde van las cosas.
* Es seguro, con herramientas integradas para la validación de entradas, protección CSRF, protección de manipulación de formularios y prevención de inyección SQL y un código base auditado a través del programa Mozilla Secure Open Source..
* Todo esto está respaldado por una licencia del MIT para uso comercial
* Cuenta con una amplia comunidad de desarrolladores quienes contribuyen al constante mejoramiento de la herramienta y ofrecen soporte a sus usuarios.

Algunas de las empresas que usan este framework son: Hyundai, Calzedonia y Billabong.

Se realizó el mismo proceso de buscar casos de uso relevantes de las herramientas de Spring Boot y Angular, sin embargo, no se encontró información sobre estos a nivel detallado más que menciones respecto al primero que se usa en Netflix y Amazon y al segundo usado en Google y Nike.

En cuanto a las estadísticas, Angular se ubicó en el top 3 en 2021 en términos de desarrollo web. Según una encuesta de StackOverflow, Angular ocupó el tercer lugar con un 25,1% de uso, y en Github recibió entre 60.000 y 80.000 estrellas. Además, se encontraron entre 10.000 y 15.000 ofertas de trabajo en busca de esta herramienta en Estados Unidos.

Por otro lado, Spring Boot no apareció en los rankings encontrados, pero Spring, una herramienta similar, se posicionó en el cuarto lugar después de Angular. Para diciembre de 2020, Spring contaba con entre 20.000 y 40.000 estrellas en Github y el 16,4% de los encuestados por StackOverflow lo utilizaban. Representaba el tercer framework más solicitado en Estados Unidos, con entre 10.000 y 15.000 ofertas de trabajo.

Finalmente, MaríaDB es la segunda base de datos más apetecida por los usuarios Open Source gracias a su rendimiento y facilidad de uso. En comparación otras bases de datos y de acuerdo a la encuesta de StackOverflow, la utilizaban el 16.8% de los usuarios.

## **Ejemplo práctico y funcional relacionando los temas**

Enlace al repositorio del FrontEnd:­­­­­

<https://github.com/AlejoDiazLote/FrontArquiTallerIIGrupo1.git>

Enlace al repositorio del BackEnd:

11<https://github.com/AlejoDiazLote/BackArquiTallerIIGrupo1.git>

Enlace al repositorio de las Bases de Datos:

<https://github.com/AlejoDiazLote/BaseDatosArquiTallerIIGrupo1.git>

## **Conclusiones**

* La combinación de MVC, Angular y Spring Boot es una excelente opción para el desarrollo de aplicaciones web altamente escalables, eficientes y modulares.
* Con Spring Boot se pueden crear aplicaciones que sean altamente escalables y que puedan manejar una gran cantidad de usuarios y datos.
* La arquitectura de software MVC separa la lógica de presentación de la lógica de negocio y la gestión de datos, lo que proporciona una mayor modularidad y escalabilidad en el desarrollo de aplicaciones.

## **Lecciones aprendidas**

Por una parte, Angular y Spring Boot son herramientas de desarrollo web ampliamente utilizadas y populares en la actualidad. Angular es un framework de desarrollo de aplicaciones web de código abierto, respaldado por Google, que se utiliza comúnmente para crear aplicaciones de alta calidad y de una sola página. Por otro lado, Spring Boot es un framework de desarrollo de aplicaciones empresariales de código abierto, que se utiliza para crear aplicaciones empresariales escalables.

Además, ambas herramientas son compatibles con varias bases de datos, como MariaDB, lo que las hace aún más valiosas en el mercado laboral. MariaDB es una base de datos de código abierto y uno de los sistemas de gestión de bases de datos más populares en la actualidad. Su compatibilidad con Angular y Spring Boot permite a los desarrolladores trabajar con una base de datos segura, confiable y fácil de usar.

Por otra parte y respecto al patrón asignado, el patrón de diseño MVC es ampliamente utilizado en el desarrollo de aplicaciones web debido a sus beneficios en la organización, escalabilidad y mantenibilidad del código. La separación clara de las responsabilidades en la lógica de la aplicación, la presentación y la gestión de datos proporciona una estructura clara y modular que facilita la creación y el mantenimiento de aplicaciones web complejas. Además, al permitir la separación de los componentes, el patrón MVC también facilita la colaboración entre los miembros del equipo de desarrollo

## **Referencias**

* 1. <https://folk.universitetetioslo.no/trygver/2007/MVC_Originals.pdf>
  2. <https://www.bbvaapimarket.com/es/mundo-api/mvc-desarrollo-fiable-de-aplicaciones-web/>
  3. <https://medium.com/bumble-tech/do-mvc-like-its-1979-da62304f6568>
  4. <https://cakephp.org/>
  5. <http://catalyst.perl.org/#sites>
  6. <https://rubyonrails.org/>
  7. <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>
  8. <https://angular.io/>
  9. <https://spring.io/projects/spring-boot>
  10. <https://www.bbva.com/es/patrones-de-diseno-en-el-desarrollo-de-software/>
  11. <https://www.amazon.com/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612>
  12. <https://www.amazon.com/Patterns-Enterprise-Application-Architecture-Martin/dp/0321127420>
  13. <https://www.amazon.com/Head-First-Design-Patterns-Brain-Friendly/dp/0596007124>
  14. <https://mariadb.com>
  15. <https://www.udemy.com/course/spring-boot-tutorial-for-beginners/>
  16. <https://docs.spring.io/spring-framework/docs/current/reference/html/>
  17. <https://spring.io/projects/spring-boot>
  18. <https://www.calibraint.com/blog/the-evolution-of-angular-versions>
  19. <https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm>
  20. <https://dzone.com/articles/history-of-spring-framework-spring-boot-framework>
  21. <https://mariadb.org/es/>
  22. <https://blog.hubspot.es/website/que-es-java>
  23. <https://platzi.com/clases/1798-javascript-navegador/25680-historia-de-javascript/>
  24. <https://insights.stackoverflow.com/survey/2020>
  25. <https://towardsdatascience.com/top-10-in-demand-web-development-frameworks-in-2021-8a5b668be0d6>
  26. <https://www.peerspot.com/categories/open-source-databases>
  27. <https://www.peerspot.com/categories/relational-databases-tools>