

**DISEÑO Y DESARROLLO DE UNA PLATAFORMA DE
APRENDIZAJE EN LINEA**

ALEJANDRO VELEZ ARBOLEDA

**Trabajo de grado para optar al título de Tecnólogo en desarrollo de
software**

Asesor

Liliana maría García Aguirre

Magister en Ingeniería de Software

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO

FACULTAD DE INGENIERÍA

TECNOLOGÍA EN DESARROLLO DE SOFTWARE

MEDELLIN

2023

**DISEÑO Y DESARROLLO DE UNA PLATAFORMA DE
APRENDIZAJE EN LINEA**

ALEJANDRO VELEZ ARBOLEDA

**Trabajo de grado para optar al título de Tecnólogo en desarrollo de
software**

Asesor

Liliana maría García Aguirre

Magister en Ingeniería de Software

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO

FACULTAD DE INGENIERÍA

TECNOLOGÍA EN DESARROLLO DE SOFTWARE

MEDELLIN

2023

Contenido

Resumen.....	6
Abstract.....	7
Glosario.....	8
Introducción.....	9
1. Planteamiento del problema.....	10
1.1. Descripción.....	10
1.2. Formulación.....	10
2. Justificación.....	11
3. Objetivos.....	12
3.1. Objetivo general.....	12
3.2. Objetivos específicos.....	12
4. Marco teórico.....	13
4.1 Bases teóricas.....	13
4.1.2 Educación en Línea y Tecnología Educativa.....	13
4.2 Antecedentes.....	13
5. Metodología.....	14
5.1 Tipo de proyecto.....	14
5.2 Diseño del Software.....	14
5.3 Tecnologías y Herramientas.....	16
6. Resultados.....	18
6.1 Planeación.....	18
6.1.1 Historias de Usuario.....	18
6.1.2 Backlog de Historias de usuario.....	22
6.1.2 Diagrama de Casos de Uso.....	23
6.2 Diseño.....	23
6.2.1 Arquitectura del Proyecto - Modelo Vista Controlador.....	23
6.2.2 Diagrama de Clases.....	24
6.2.3 Diseño de Base de Datos - Diagrama Entidad Relación.....	24
6.2.4 Interfaz de usuario.....	25
6.4 Implementación.....	25
6.5 Pruebas.....	29
7. Conclusiones.....	30
8. Recomendaciones.....	31
9. Referencias.....	32
10. Anexos.....	33

Lista de figuras

	Pág.
Ilustración 1: Diagrama de casos de uso	23
Ilustración 2: Arquitectura del proyecto- Modelo vista controlador.....	23
Ilustración 3: Diagrama de clases	24
Ilustración 4: Diagrama entidad relación	24
Ilustración 5: Interfaz de usuario	25
Ilustración 6: Modulo servicio	26
Ilustración 7: Modulo implementación	26
Ilustración 8: Modulo vista	27
Ilustración 9: Modulo controlador	28
Ilustración 10: Pruebas unitarias I.....	29
Ilustración 11: Pruebas Unitarias II	29

Lista de tablas

	Pág.
Tabla 1: HU Acceso a registro y entrada a la plataforma	18
Tabla 2: HU Acceso a contenido de cursos	18
Tabla 3: HU Actualizar datos personales.....	18
Tabla 4: HU Visualizar cursos propios	19
Tabla 5: HU Actualización de cursos propios	19
Tabla 6: HU Creación de cursos propios.	19
Tabla 7: HU Actualización de perfil de instructor	20
Tabla 8: HU Visualización de Cursos.....	20
Tabla 9: HU Actualización de cursos	20
Tabla 10: HU Eliminación de cursos	21
Tabla 11: HU Creación de cursos	21
Tabla 12: HU Visualización de instructores	21
Tabla 13: HU Búsqueda de instructores	22

Resumen

Este proyecto se centra en el desarrollo de una plataforma de aprendizaje en línea con el propósito de mejorar la experiencia educativa. La plataforma busca ser intuitiva y de fácil uso. Ante la creciente demanda de métodos educativos flexibles en Colombia, esta plataforma se pondría como una respuesta a los desafíos que surgen con la enseñanza y el aprendizaje a distancia. Su enfoque en la gratuidad y la simplificación del acceso tiene como objetivo promover la educación en línea y beneficiar a la comunidad educativa en su conjunto.

Para abordar este proyecto, se adopta la metodología Scrum debido a que se tomó historias de usuario y requisitos de la aplicación para construirla además Scrum promovía un enfoque ágil en el proceso de desarrollo, ayudando a llevar a cabo una planificación que abarcará desde la definición de requisitos hasta la realización de pruebas.

La plataforma se diseñó siguiendo una estructura organizada se adoptó una arquitectura basada en el patrón Modelo-Vista-Controlador (MVC), seguido se analizó las necesidades de maestros, estudiantes y administradores para definir las funcionalidades clave de la plataforma. incluyendo la estructura de bases de datos y la interfaz de usuario, para una experiencia intuitiva y eficiente.

El proyecto se organiza en torno a historias de usuario, con roles específicos para estudiantes, instructores y administradores y se programó las funcionalidades correspondientes de cada rol. las historias de usuario se priorizaron y se hicieron en términos de esfuerzo. Se hizo una evaluación a través de pruebas de usabilidad y prueba unitarias en las implementaciones.

Abstract

This project focuses on the development of an online learning platform with the purpose of enhancing the educational experience. The platform aims to be intuitive and easy to use. In response to the growing demand for flexible educational methods in Colombia, this platform is positioned as a solution to the challenges that arise with distance teaching and learning. Its emphasis on free access and simplification aims to promote online education and benefit the entire educational community.

To address this project, the Scrum methodology was adopted. User stories and application requirements were gathered to build the platform, and Scrum promoted an agile approach in the development process, facilitating planning from requirement definition to testing.

The platform was designed following an organized structure, adopting an architecture based on the Model-View-Controller (MVC) pattern. The needs of teachers, students, and administrators were analyzed to define key platform functionalities, including database structure and user interface, for an intuitive and efficient experience.

The project is organized around user stories, with specific roles for students, instructors, and administrators. Corresponding functionalities for each role were programmed, and user stories were prioritized based on effort. Evaluation was conducted through usability tests and unit testing in the implementations.

Glosario

HTML (Hypertext Markup Languaje): Es un lenguaje de marcado que nos permite crear sitios o cualquier aplicación web.

Implementación: Se refiere a la parte específica de una clase que define como se realizan las operaciones o métodos declarados en la interfaz.

Interfaz: Se refiere a un conjunto de métodos que se declaran y en donde las clases que implementan esa interfaz debe cumplir.

Módulos: Son partes individuales de un programa o sistema que realizan funciones específicas. Ayudan a organizar y dividir el código en partes.

MVC(Modelo-Vista-Controlador): Es un patrón de diseño de software que divide la aplicación en tres componentes: Modelo (Gestión de datos), Vista (Interfaz de usuario), Controlador (Lógica de control).

Scrum: Es una metodología ágil utilizada en proyectos mas que todo en el desarrollo de software.

Sprint: En el contexto de scrum los sprints son periodos de tiempo cortos, en los que se planifica, desarrollan y se integran partes del proyecto.

Thymeleaf: Es un motor de plantillas que se utiliza en HTML para comunicarnos entre el frontend y el backend de manera sencilla y eficiente.

Introducción

La educación en línea se ha convertido en una parte esencial del proceso educativo, acelerada aún más por el cambio constante de la tecnología en el mundo. El propósito fundamental de esta iniciativa es proporcionar una plataforma de aprendizaje accesible, amigable e intuitiva, diseñada para abordar la creciente demanda de métodos de enseñanza eficaces y flexibles en el contexto actual.

La plataforma surge como una respuesta a la creciente demanda de métodos educativos flexibles accesibles que permitan el aprendizaje a distancia, buscando mejorar la accesibilidad, la usabilidad y la experiencia del usuario. La necesidad de plataformas efectivas en el entorno digital es crucial, ya que la eficacia de las plataformas de aprendizaje en línea está fuertemente ligada a la satisfacción de los estudiantes y el impacto en su proceso educativo.

El desarrollo de esta plataforma sigue un enfoque organizado y estructurado, utilizando la metodología Scrum para identificar necesidades, validar el funcionamiento y lograr una colaboración constante. A lo largo del proceso, abordará la definición de requisitos, la creación de historias de usuario y la realización de pruebas. El uso de Scrum fomentará la flexibilidad y la eficiencia en el desarrollo del proyecto.

Este proyecto tiene como objetivo enriquecer la experiencia del estudiante, beneficiar a la comunidad educativa en general. A través de la personalización del aprendizaje y la interacción fluida, se espera que este proyecto contribuya al avance continuo del proceso educativo en el contexto digital actual. Además, se promueve el uso de herramientas tecnológicas para mejorar la experiencia educativa.

El proyecto se desarrolla utilizando tecnologías clave como Java, el Framework Spring, HTML, CSS, Thymeleaf y Bootstrap, lo que garantiza una base sólida para el desarrollo de la plataforma. Con la finalización de este proyecto.

1. Planteamiento del problema

1.1. Descripción

Esta plataforma busca responder a la creciente demanda de métodos educativos flexibles y eficientes en Colombia. La educación en línea por estos tiempos ha estado en auge, y este proyecto se propone mejorar la accesibilidad, usabilidad y experiencia del usuario.

Según un estudio de la red Ilumino (Republica, 2023), para 2018 solo el 10 % de los estudiantes en Colombia se encontraban estudiando en modalidad virtual. No obstante, para la pandemia aumentaron a 70% por lo cual tuvo un efecto positivo para la industria de aprendizaje a distancia no solo para estudiante si no para adultos. Aprender en línea tienen grandes ventajas para los niños y padres ya que trae flexibilidad de horarios integración de recursos interactivos y monitorear el progreso de aprendizaje de los estudiantes o ya sean adultos.

Sin embargo, esta plataforma de aprendizaje en línea está posicionada de manera que aborde los desafíos que surgen a raíz de la creciente demanda de métodos educativos flexibles y accesibles que permitan la enseñanza y el aprendizaje a distancia. Con su enfoque en la gratuidad, la simplificación del acceso y la amplia disponibilidad para quienes la necesiten.

1.2. Formulación

¿Cómo mejorar la usabilidad de una plataforma educativa en línea para estudiantes e instructores de educación superior, a fin de lograr una experiencia de aprendizaje más efectiva y satisfactoria?

2. Justificación

En el contexto actual de la educación en Colombia, la incorporación de la tecnología desempeña un papel fundamental en la transformación del acceso y la entrega del conocimiento. La demanda de métodos de aprendizaje en línea eficaces y flexibles ha experimentado un aumento significativo tanto por parte de los estudiantes como de los instructores. Sin embargo, la efectividad de las plataformas educativas en línea está vinculada a su usabilidad, y a la interacción fluida entre los diversos actores involucrados en el proceso educativo, la creación de cursos online y enseñanza a distancia con una metodología poco cuidada y sin estructura es sinónimo de fracaso (Economista, 2020).

Este proyecto surge como respuesta a esta necesidad, desarrollar una Plataforma Educativa que potencie una experiencia de aprendizaje y centrada en el usuario. La plataforma permitirá a los instructores crear y compartir cursos de manera eficiente, ofreciendo contenido que se ajuste a las necesidades específicas de los estudiantes. En paralelo, los estudiantes podrán inscribirse en cursos de su elección y acceder a los recursos de manera intuitiva, simplificando su proceso de aprendizaje.

El propósito fundamental de este proyecto radica en enriquecer la experiencia del estudiante y que se adapte a las cambiantes necesidades de la comunidad educativa, la personalización del aprendizaje y una interacción fluida contribuirán de manera sustancial al avance continuo del proceso educativo en el contexto digital actual. A nivel social, la plataforma tiene el potencial de enriquecer el acceso a la educación, empoderando a los estudiantes y facilitando la labor de los instructores. En términos tecnológicos, se promueve la adopción de herramientas para facilitar la experiencia educativa. En cuanto al aspecto ambiental, la plataforma promueve la reducción de recursos físicos al ofrecer un entorno virtual de aprendizaje. En resumen, este proyecto busca una transformación positiva en la educación en línea, aportando a múltiples áreas y beneficiando a la comunidad educativa en su conjunto.

3. Objetivos

3.1. Objetivo general

Desarrollar una plataforma de aprendizaje en línea que mejore la experiencia educativa tanto para maestros como para estudiantes, que proporcione una interfaz intuitiva que permita la creación, distribución y acceso eficiente a cursos.

3.2. Objetivos específicos

- Analizar las necesidades de maestros, estudiantes y administradores para definir las funcionalidades clave y características de la plataforma de aprendizaje en línea.
- Diseñar la arquitectura de la plataforma, incluyendo la estructura de bases de datos, la interfaz de usuario y los flujos de navegación, para una experiencia intuitiva y eficiente.
- Programar las funcionalidades esenciales de la plataforma, como la creación y gestión de cursos, inscripción de estudiantes.
- Implementar un sistema de autenticación seguro y eficaz, que utilice la automatización para generar credenciales de acceso según el rol de cada usuario registrado.
- Evaluar la plataforma a través de pruebas de usabilidad y pruebas de usuario, asegurando su funcionamiento estable, la seguridad de los datos y la satisfacción tanto de maestros como de estudiantes.

4. Marco teórico

4.1 Bases teóricas

El marco teórico de este proyecto se fundamenta en diversas áreas de conocimiento que convergen para respaldar y enriquecer la propuesta del desarrollo de la plataforma.

4.1.2 Educación en Línea y Tecnología Educativa

La educación en línea ha transformado la forma en que las personas acceden al conocimiento y adquieren habilidades. La tecnología educativa se ha convertido en una fuente clave para brindar oportunidades de aprendizaje flexible y a distancia. Diversos enfoques se han adaptado al contexto digital. Estos enfoques influyen en el diseño y la estructura de la plataforma de aprendizaje en línea (GCF Global, 2023).

Diseño y Desarrollo de Plataformas en Línea:

El diseño centrado en el usuario es fundamental para garantizar que la plataforma sea intuitiva y fácil de usar. Los principios de usabilidad y accesibilidad guiarán la creación de una interfaz de usuario que facilite la navegación y la interacción. La elección de tecnologías web modernas y la arquitectura de bases de datos eficiente respaldarán la funcionalidad y de la plataforma.

Gestión de Contenidos y Acceso:

La gestión eficiente de contenidos se basa en la organización y presentación estructurada de materiales educativos. La automatización de la administración de usuarios y la distribución de contenidos garantizará un acceso fluido y oportuno a los recursos educativos, mejorando la experiencia de aprendizaje.

Justificación y Significancia:

La relevancia de este proyecto radica en la creciente demanda de educación en línea y en la necesidad de mejorar la experiencia educativa en entornos digitales. La personalización tiene el potencial de impactar positivamente la calidad del aprendizaje y la eficiencia en la gestión de la educación en línea, contribuyendo a la formación de individuos más competentes y capacitados en la sociedad actual.

4.2 Antecedentes

La constante evolución de la tecnología educativa, resulta esencial comprender las dinámicas y desafíos asociados a la implementación de la plataforma de enseñanza virtual en entornos educativos. La creación o selección de estas plataformas requiere una cuidadosa consideración de sus características, herramientas y estándares (SánchezRodríguez, 2009).

WebCT: Es una plataforma tecnológica educativa con fines educativos que integra herramientas para la administración, comunicación del aprendizaje para la web.

Ofrecía ciertas ventajas que incluía una interfaz de usuario intuitiva, los profesores valoraban su capacidad de la integración de contenido multimedia para mejorar la experiencia de aprendizaje. Sin embargo, había desventajas como los costos, lo que podría ser un obstáculo para instituciones con presupuestos limitados.

5. Metodología

5.1 Tipo de proyecto

El proyecto se centrará en diseñar, desarrollar e implementar una plataforma aprendizaje en línea bajo la metodología ágil SCRUM.

Metodología SCRUM:

SCRUM es un enfoque ágil de gestión de proyectos que se utiliza comúnmente en el desarrollo de software, pero que también puede aplicarse a una variedad de otros proyectos. Se basa en principios de colaboración, adaptación continua y entrega iterativa (Hurtado, 2021). A continuación, se destacan algunos aspectos clave de SCRUM:

Roles Clave: En SCRUM, hay tres roles principales: el Scrum Master, el Product Owner y el Equipo de Desarrollo. Cada uno tiene responsabilidades específicas para garantizar que el proyecto avance de manera efectiva.

Sprints: SCRUM divide el proyecto en ciclos de desarrollo llamados "sprints", que suelen tener una duración fija (por ejemplo, 2-4 semanas). Durante cada sprint, el equipo se enfoca en la entrega de un conjunto específico de características o funcionalidades.

Product Backlog: El Product Owner mantiene una lista priorizada de elementos de trabajo llamada "Product Backlog", que contiene todas las características, mejoras y correcciones necesarias para el proyecto.

Iteración y Mejora Continua: SCRUM promueve la iteración constante y la mejora continua del producto y del proceso. Después de cada sprint, se realiza una retrospectiva para identificar áreas de mejora.

Entrega Incremental: SCRUM se basa en la entrega iterativa, lo que significa que puede entregar partes funcionales de la plataforma en sprints cortos. Esto permite obtener retroalimentación temprana y ajustar el producto según sea necesario.

Adaptabilidad: Si el proyecto enfrenta cambios en los requisitos o necesidades a lo largo del tiempo, SCRUM te permite adaptarte de manera efectiva. Puedes ajustar el Product Backlog y priorizar las características según las necesidades cambiantes.

Mejora Continua: SCRUM incluye la práctica de la retrospectiva del sprint, lo que te permite identificar y abordar problemas y desafíos en el proceso de desarrollo de manera regular.

5.2 Diseño del Software

Se plantea una estructura que garantice la eficiencia de la Plataforma. El diseño se basa en la modularización de componentes y la interacción fluida entre ellos bajo la arquitectura MVC (Model-View-Controller) para que el proyecto sea mucho más manejable ante cualquier mejora o cambio.

MVC (Model-View-Controller)

Es una arquitectura de software utilizada comúnmente en el diseño y desarrollo de aplicaciones, especialmente en aplicaciones web y de software. El propósito principal del patrón MVC es separar la aplicación en tres componentes interconectados para facilitar la gestión, mantenimiento y escalabilidad del software (Mozilla, 2023). Aquí está una breve descripción de cada componente:

Modelo (Model): El Modelo representa los datos y la lógica de negocio de la aplicación. Es responsable de acceder, procesar y almacenar los datos. En una aplicación web, esto podría implicar recuperar datos de una base de datos, realizar cálculos y aplicar lógica empresarial.

Vista (View): La Vista se encarga de la presentación de los datos al usuario. Es la interfaz de usuario que muestra la información del Modelo de una manera comprensible. En el contexto web, esto podría ser una página HTML, una plantilla, o una interfaz de usuario generada dinámicamente.

Controlador (Controller): El Controlador actúa como intermediario entre el Modelo y la Vista. Recibe las solicitudes del usuario, procesa la entrada, interactúa con el Modelo para obtener los datos necesarios y actualiza la Vista en consecuencia. El Controlador controla el flujo de la aplicación y toma decisiones basadas en las acciones del usuario.

División en Módulos:

La arquitectura del software se organizará en varios módulos interconectados, cada uno con una función específica y bien definida. Estos módulos incluirán:

Módulo de Autenticación y Seguridad: Responsable de gestionar el proceso de inicio de sesión y autenticación de los usuarios, asegurando un acceso seguro a la plataforma.

Módulo de Administración: Permitirá a los administradores gestionar usuarios, cursos y contenido.

Módulo de Instructores: Ofrecerá a los instructores las herramientas necesarias para crear, gestionar actualizar cursos de manera eficiente.

Módulo de Estudiantes: Proporcionará a los estudiantes la capacidad de explorar, inscribirse y acceder a los cursos.

Interacción entre Componentes: Los módulos se comunicarán entre sí a través de interfaces. Por ejemplo, el módulo de Autenticación y Seguridad proporcionará información de autenticación a los otros módulos después de la validación exitosa del usuario. El módulo de Administración interactuará con los módulos de Instructores y Estudiantes para gestionar inscripciones.

Manejo de Datos: La base de datos será un componente fundamental para almacenar y administrar datos como usuarios, cursos, contenido datos personales de los usuarios. Se empleará una base de datos relacional bien estructurada para garantizar la integridad de los datos y permitir consultas eficientes. El diseño de la base de datos considerará la relación entre tablas y la optimización para búsquedas y actualizaciones.

5.3 Tecnologías y Herramientas

Java: Lenguaje de programación

Es ampliamente utilizado en el desarrollo de aplicaciones de software. Java es un lenguaje de programación de alto nivel, orientado a objetos, con características que promueven la portabilidad y la seguridad en las aplicaciones". Java se distingue por su versatilidad y robustez, lo que lo hace adecuado para una variedad de aplicaciones, desde aplicaciones de escritorio hasta sistemas embebidos y aplicaciones web (Aws, 2023).

Java ofrece varias características clave que lo convierten en una elección sólida para proyectos de desarrollo de software, como la plataforma de este proyecto:

Portabilidad: Java se ejecuta en la Máquina Virtual Java (JVM), lo que le permite ser independiente de la plataforma. Esto significa que el código Java puede ejecutarse en diferentes sistemas operativos sin necesidad de modificaciones significativas.

Orientación a Objetos: Java es un lenguaje orientado a objetos, lo que facilita la organización y estructuración del código en objetos reutilizables. Esto es especialmente útil para proyectos complejos como plataformas educativas en línea.

Seguridad: Java ofrece un entorno de ejecución seguro a través de su sistema de administración de memoria y la capacidad de restringir el acceso a recursos del sistema. Esto es crucial para garantizar la seguridad de los datos y la integridad del sistema.

Amplia Biblioteca Estándar: Java proporciona una amplia biblioteca estándar con clases y métodos predefinidos que simplifican el desarrollo de aplicaciones. Esto acelera el proceso de desarrollo y permite centrarse en la funcionalidad específica.

Comunidad Activa: Java cuenta con una comunidad activa de desarrolladores y una gran cantidad de recursos en línea, como foros y tutoriales, que facilitan el aprendizaje y la resolución de problemas.

Spring Framework

El Framework de Spring es una herramienta poderosa y versátil utilizada en el desarrollo de aplicaciones empresariales y web en Java. Según (Walls, 2021)"Spring es un framework de desarrollo de aplicaciones que facilita la creación de aplicaciones Java de alta calidad". Spring se caracteriza por su enfoque en el modularidad, la escalabilidad y la facilitación del desarrollo ágil.

Inyección de Dependencias: Spring facilita la gestión de componentes y objetos al ofrecer un sistema de inyección de dependencias. Esto promueve la reutilización de código y la separación de preocupaciones, lo que simplifica el mantenimiento y la evolución del sistema.

Abstracción de Tecnologías: Spring proporciona abstracciones sobre tecnologías comunes, como JDBC y JMS, lo que simplifica el acceso a bases de datos y la integración de sistemas. Esto acelera el proceso de desarrollo y reduce la complejidad.

Aspect-Oriented Programming (AOP): Spring facilita la implementación de AOP, lo que permite modularizar y gestionar preocupaciones transversales, como la seguridad y el registro, de manera más eficiente.

Módulos y Extensiones: Spring se compone de varios módulos que se pueden utilizar según las necesidades específicas del proyecto. Además, ofrece extensiones para integrar tecnologías como Spring Security y Spring MVC.

Gestión de Transacciones: Spring ofrece un soporte robusto para la gestión de transacciones, lo que es fundamental para garantizar la integridad de los datos en aplicaciones empresariales.

HTML (HyperText Markup Language)

Hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El libro *Introducción a HTML y JavaScript* de (Gomez, 2000) cita que “el lenguaje HTML es un formato muy simple para crear documentos de hipertexto que pueden ser visualizados en múltiples plataformas. Así al dar cierto formato a la información, nos aseguramos de que cualquier usuario de una computadora personal pueda verla, incluyendo elementos como imágenes, audio, video, e incluso programas completos.”.

BOOTSTRAP

Bootstrap es un marco de diseño front-end que facilita la creación de sitios web atractivos y responsivos. Proporciona una serie de estilos predefinidos y componentes listos para usar que permiten a los desarrolladores diseñar páginas web de manera eficiente y profesional (Bootstrap, 2023).

En lugar de comenzar desde cero con el diseño y la estructura de una página, Bootstrap ofrece una serie de clases y elementos que se pueden aplicar para lograr rápidamente un diseño coherente y moderno.

Thymeleaf

Thymeleaf es un motor de plantillas de código abierto diseñado para la creación de vistas dinámicas en aplicaciones web, especialmente en el contexto del lenguaje de programación Java. Permite generar contenido web dinámico al combinar plantillas HTML con datos provenientes del servidor y proporciona una forma sencilla de crear páginas web que se adaptan y actualizan en función de la información que se procesa en el lado del servidor (Fernández, 2018)

Thymeleaf fue creado por Daniel Fernández en 2011. Su objetivo principal era proporcionar una alternativa eficiente y fácil de usar a otras tecnologías de plantillas, como JSP (JavaServer Pages) y FreeMarker, para el desarrollo de aplicaciones web en Java. Thymeleaf se ha vuelto muy popular en la comunidad de desarrollo Java debido a su sintaxis sencilla y a su enfoque en la creación de páginas web mantenibles y adaptables.

6. Resultados

6.1 Planeación

6.1.1 Historias de Usuario

Historias de Usuario Estudiante:

Tabla 1: HU Acceso a registro y entrada a la plataforma

Rol: Estudiante	
Historia de Usuario:	Acceso a registro y entrada a la plataforma
Como:	Estudiante
Quiero:	Poder registrarme a la plataforma
Para	Inscribirme a la variedad de cursos disponibles.
Criterios de Aceptación:	Como estudiante debo poder registrarme y ingresar a la plataforma

Tabla 2: HU Acceso a contenido de cursos

Rol: Estudiante	
Historia de Usuario:	Acceso a Contenido de Cursos
Como:	Estudiante
Quiero:	Poder acceder al contenido de los cursos en los que estoy inscrito.
Para	Para Ampliar mi conjunto de cursos disponibles.
Criterios de Aceptación:	En mi perfil, debo ver una lista de los cursos en los que estoy inscrito.

Tabla 3: HU Actualizar datos personales

Rol: Estudiante	
Historia de Usuario:	Actualizar datos personales
Como:	Estudiante
Quiero:	Poder actualizar mi perfil de estudiante
Para:	Mantener mi información actualizada y relevante
Criterios de Aceptación:	En mi perfil tendrá la opción de actualizar información donde me permitirá modificarla.

Rol: Instructor

Tabla 4: HU Visualizar cursos propios

Rol: Instructor	
Historia de Usuario:	Visualizar cursos propios
Como:	Instructor
Quiero:	Quiero Ver la lista de los cursos que estoy impartiendo
Para:	Para Mantener un seguimiento de los cursos que gestiono
Criterios de Aceptación:	Se muestra una lista de cursos con sus nombres y detalles básicos, que son impartidos por el instructor.

Tabla 5: HU Actualización de cursos propios

Rol: Instructor	
Historia de Usuario:	Actualización de Cursos Propios
Como:	Instructor
Quiero:	Poder actualizar la información de un curso que imparto
Para:	Mantener la información actualizada y relevante para los estudiantes
Criterios de Aceptación:	Puedo acceder a la página de actualización del curso.

Tabla 6: HU Creación de cursos propios.

Rol: Instructor	
Historia de Usuario:	Creación de Cursos Propios.
Como:	Instructor
Quiero:	Poder crear nuevos cursos y agregar información relevante.
Para:	Para Ofrecer nuevos cursos a los estudiantes.
Criterios de Aceptación:	Puedo acceder a la página donde se crearán los cursos.

Tabla 7: HU Actualización de perfil de instructor

Rol: Instructor	
Historia de Usuario:	Actualización de Perfil de Instructor.
Como:	Instructor
Quiero:	Poder actualizar mi perfil como instructor
Para:	Mantener mi información personal actualizada para los estudiantes
Criterios de Aceptación:	En mi perfil de instructor tendré la opción de actualizar información donde me permitirá modificarla.

Rol: Administrador

Tabla 8: HU Visualización de Cursos

Rol: Administrador	
Historia de Usuario:	Visualización de Cursos
Como:	Administrador
Quiero:	Quiero ver una lista de todos los cursos disponibles
Para:	Para elegir los cursos que deseo administrar y actualizar
Criterios de Aceptación:	Se muestra una lista completa de cursos con sus nombres y detalles básicos.

Tabla 9: HU Actualización de cursos

Rol: Administrador	
Historia de Usuario:	Actualización de Cursos
Como:	Administrador
Quiero:	Poder actualizar la información de un curso existente
Para:	Mantener la información de los cursos precisa y actualizada
Criterios de Aceptación:	Puedo acceder a la página de actualización del curso.

Tabla 10: HU Eliminación de cursos

Rol: Administrador	
Historia de Usuario:	Eliminación de Cursos
Como:	Administrador
Quiero:	Quiero poder eliminar un curso existente
Para:	Para eliminar cursos que ya no son relevantes o necesarios
Criterios de Aceptación:	Puedo seleccionar un curso específico para eliminarlo.

Tabla 11: HU Creación de cursos

Rol: Administrador	
Historia de Usuario:	Creación de Cursos
Como:	Administrador
Quiero:	Poder crear nuevos cursos con información relevante
Para:	Agregar nuevos cursos a la plataforma de aprendizaje
Criterios de Aceptación:	Puedo acceder al formulario de creación de cursos.

Tabla 12: HU Visualización de instructores

Rol: Administrador	
Historia de Usuario:	Visualización de Instructores
Como:	Administrador
Quiero:	Ver una lista de todos los instructores
Para:	Identificar a los instructores y gestionar sus actividades
Criterios de Aceptación:	Se muestra una lista de instructores con sus nombres y detalles básicos.

Tabla 13: HU Búsqueda de instructores

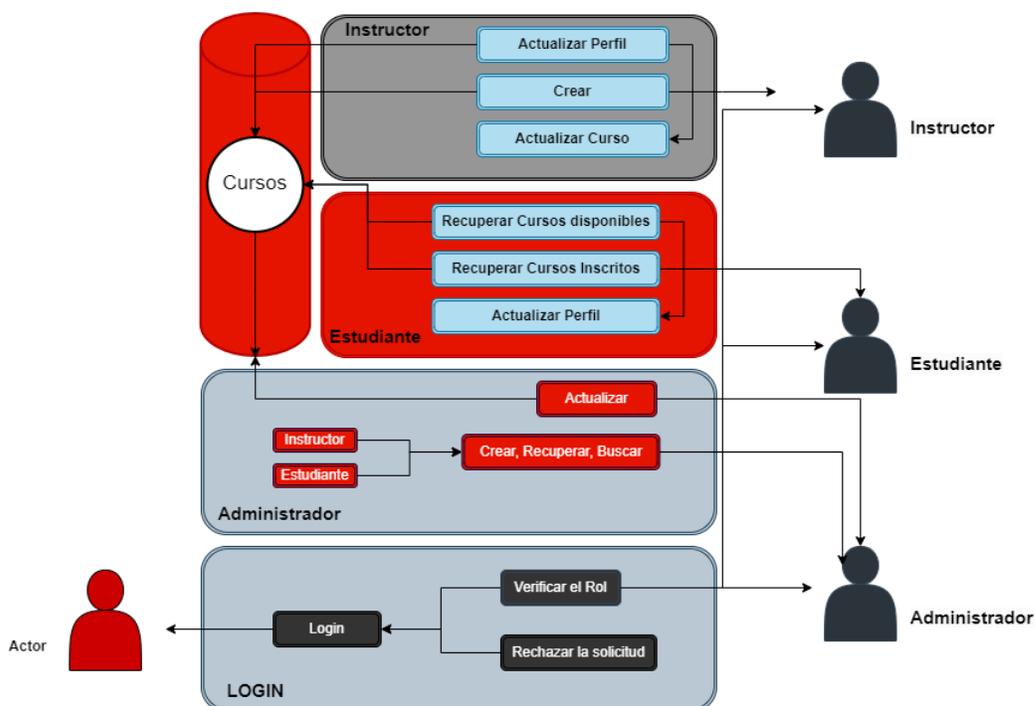
Rol: Administrador	
Historia de Usuario:	Búsqueda de Instructores
Como:	Administrador
Quiero:	Poder buscar instructores específicos por palabras clave
Para:	Encontrar rápidamente la información de instructores específicos
Criterios de Aceptación:	Puedo ingresar una palabra clave en un campo de búsqueda.

6.1.2 Backlog de Historias de usuario

Spring	Rol	Historia de Usuario	Prioridad	Estimación de Esfuerzo
Spring Uno	Instructor	Visualizar cursos propios	Media	8 puntos
	Instructor	Actualizar Curso	Baja	8 puntos
	Instructor	Crear Cursos	Alta	5 puntos
	Estudiante	Acceso a contenido de Cursos	Alta	6 puntos
	Estudiante	Búsqueda de Curso por nombre	Baja	5 puntos
Spring Dos	Estudiante	Actualizar datos personales	Baja	2 puntos
	Instructor	Actualizar Perfil	Baja	2 puntos
	Administrador	Actualizar Cursos	Baja	8 puntos
	Administrador	Crear y Eliminar Cursos	Alta	8 puntos
Spring Tres	Estudiante	Registro y entrada a la plataforma por su rol estudiante	Baja	5 puntos
	Instructor	Registro y entrada a la plataforma por su rol de instructor	Baja	5 puntos
	Administrador	Registro y entrada a la plataforma por su rol de administrador	Baja	puntos
	Administrador	Visualizar Instructores	Media	3 puntos
	Administrador	Búsqueda de instructores	Baja	5 puntos

6.1.2 Diagrama de Casos de Uso

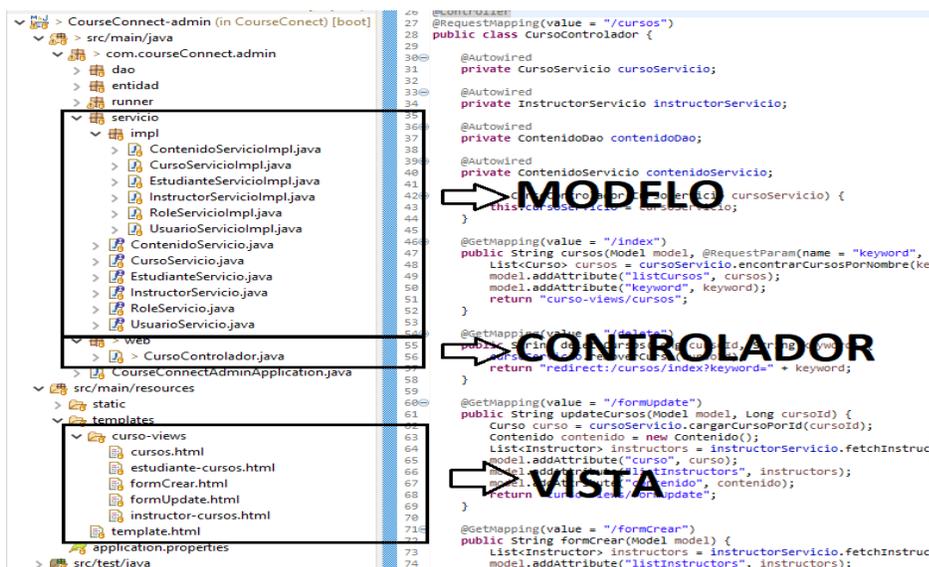
Ilustración 1: Diagrama de casos de uso



6.2 Diseño

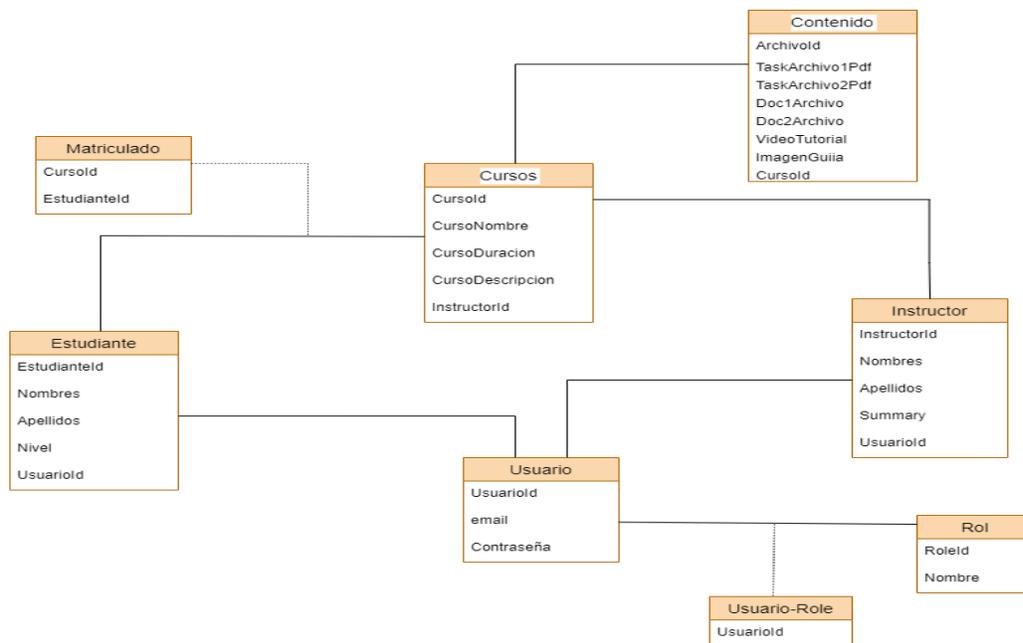
6.2.1 Arquitectura del Proyecto - Modelo Vista Controlador

Ilustración 2: Arquitectura del proyecto- Modelo vista controlador



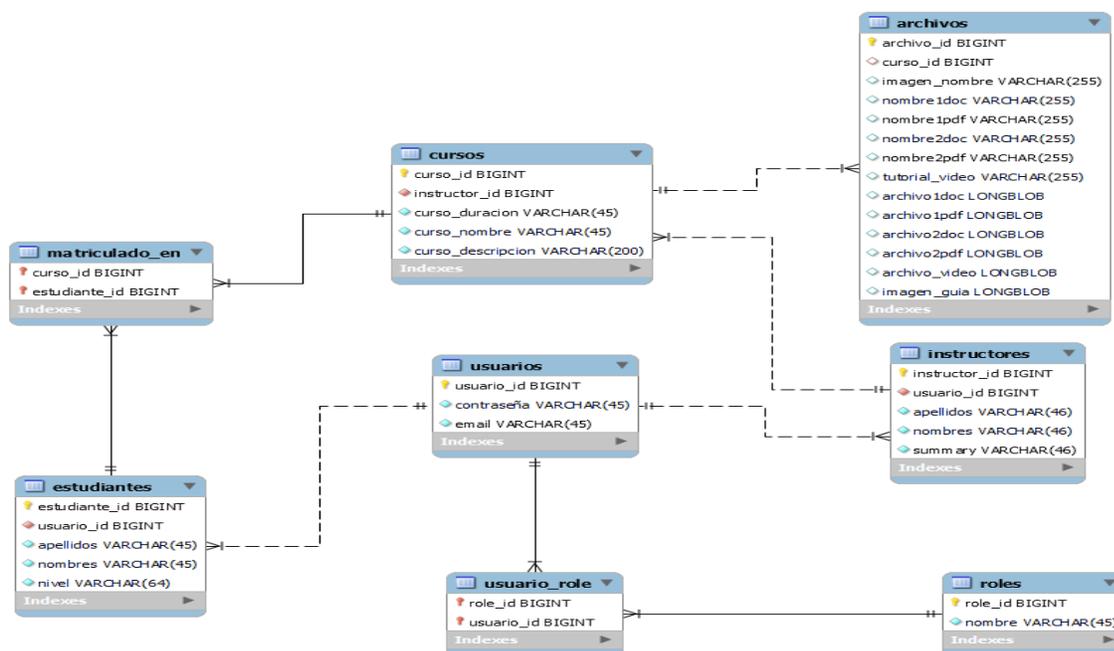
6.2.2 Diagrama de Clases

Ilustración 3: Diagrama de clases



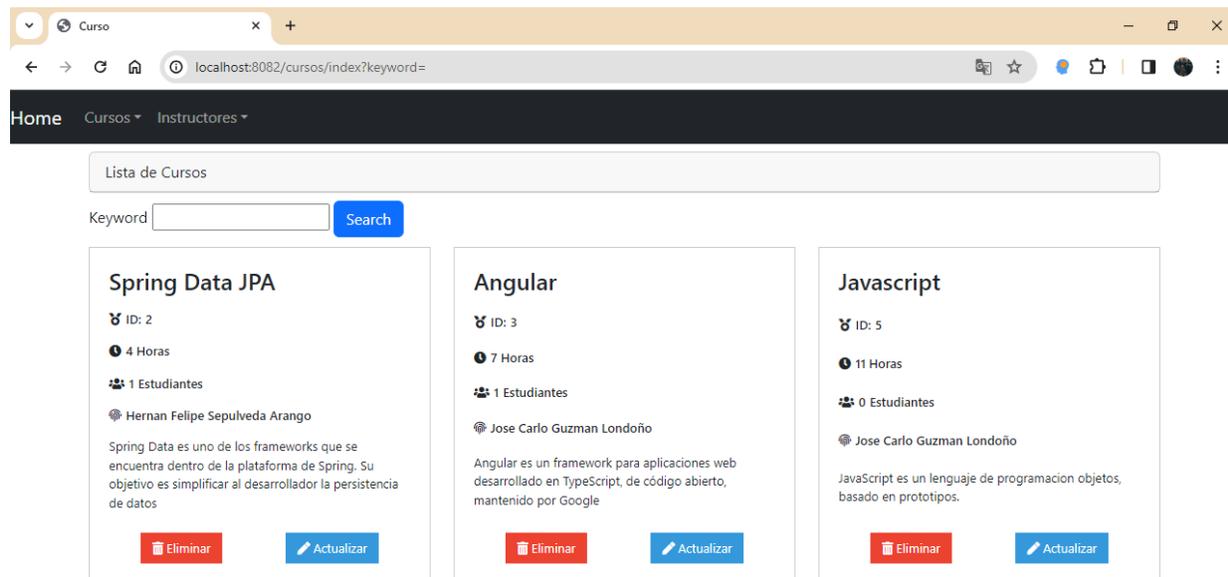
6.2.3 Diseño de Base de Datos - Diagrama Entidad Relación.

Ilustración 4: Diagrama entidad relación



6.2.4 Interfaz de usuario

Ilustración 5: Interfaz de usuario



6.4 Implementación

El desarrollo de la aplicación web se basa en el Modelo Vista Controlador (MVC) , este patrón de diseño realiza una división en tres capas o componentes distintos es decir hace una división de la forma en la que se va a representar la información y la forma en que se va a presentar la interacción del usuario con el sistema.

Se aplica el concepto de la reutilización de código y separación de conceptos de tal manera que se pueda dividir el proyecto en secciones más pequeñas que faciliten la tarea de que en el momento de realizar mantenimiento o algún tipo de cambio sea posible hacerlo de manera más rápida.

En el modelo se maneja todo lo relacionado con la información por la cual el sistema opera, es decir, este se encarga de realizar consultas, actualizaciones, establecer privilegios de acceso a información y demás. Su relación con los otros componentes básicamente es la de enviar información a las vistas y recibir solicitudes por parte del controlador.

Ilustración 6: Modulo servicio

```

1 package com.courseConnect.admin.servicio;
2
3 import java.util.List;
4
5
6
7 public interface CursoServicio {
8
9     Curso cargarCursoPorId(Long cursoId);
10
11     Curso crearCurso(String cursoNombre, String cursoDuracion, String cursoDescripcion, Long instructorId);
12
13     Curso crearOrActualizarCurso(Curso curso);
14

```

En el contexto de arquitecturas de software como el Modelo-Vista-Controlador (MVC) y en el marco de desarrollo Spring, es común utilizar interfaces junto con implementaciones con el propósito de promover la separación de preocupaciones y lograr una arquitectura más modular y fácil de mantener. Esto reduce la conexión entre componentes del sistema, lo que hace que el código sea más flexible y fácil de modificar.

Ilustración 7: Modulo implementación

```

1 package com.courseConnect.admin.servicio.impl;
2
3 import java.util.List;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 @Service
20 @Transactional
21 public class CursoServicioImpl implements CursoServicio {
22
23     @Autowired
24     private CursoDao cursoDao;
25
26     @Autowired
27     private InstructorDao instructorDao;
28
29     @Autowired
30     private EstudianteDao estudianteDao;
31
32     public CursoServicioImpl(CursoDao cursoDao, InstructorDao instructorDao, EstudianteDao estudianteDao) {
33         this.cursoDao = cursoDao;
34         this.instructorDao = instructorDao;
35         this.estudianteDao = estudianteDao;
36     }
37
38     @Override
39     public Curso cargarCursoPorId(Long cursoId) {
40         return cursoDao.findById(cursoId)
41             .orElseThrow(() -> new EntityNotFoundException("Curso con Id " + cursoId + " No Encontrado"));
42     }
43
44     @Override
45     public Curso crearCurso(String cursoNombre, String cursoDuracion, String cursoDescripcion, Long instructorId) {
46         Instructor instructor = instructorDao.findById(instructorId)
47             .orElseThrow(() -> new EntityNotFoundException("Instructor con Id " + instructorId + " No Encontrado"));
48         return cursoDao.save(new Curso(cursoNombre, cursoDuracion, cursoDescripcion, instructor));
49     }
50
51     @Override
52     public Curso crearOrActualizarCurso(Curso curso) {
53         return cursoDao.save(curso);
54     }
55

```

Las vistas se centran básicamente en mostrar al usuario de una forma estética la información o lógica del negocio. Su relación o comunicación con el modelo es que esta presenta la información que allí se tiene y mediante sus funcionalidades hacen solicitudes al controlador.

Ilustración 8: Modulo vista

```

cursos.html X
1 <!DOCTYPE html>
2 <html lang="en" xmlns:layout="http://www.ultraq.net.nz/thymeleaf/Layout" layout:decorate="template">
3
4 <head>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Curso</title>
8 <link rel="stylesheet" type="text/css" th:href="@{/css/cursoStyle.css}"/>
9 <script src="https://kit.fontawesome.com/bb9bea6807.js" crossorigin="anonymous"></script>
10 <link rel="stylesheet" type="text/css" href="/webjars/bootstrap/5.3.1/css/bootstrap.min.css">
11 </head>
12 <body>
13 <div layout:fragment="content1">
14 <div class="container mt-2">
15 <div class="card">
16 <div class="card-header">Lista de Cursos</div>
17 </div>
18 <div class="card-body mt-2">
19 <form method="get" th:action="@{/index}">
20 <label>Keyword</label>
21 <input type="text" name="keyword" th:value="{keyword}">
22 <button type="submit" class="btn btn-primary">Search</button>
23 </form>
24 <div class="productos">
25 <div class="row">
26 <div th:each="curso:${listCursos}" class="col-md-4">
27 <li class="productos-list-item">
28 <div class="contenedor-caja">
29 <h4 th:text="{curso.getCursoNombre()}"></h4>
30 <h6>
31 <i class="fa fa-medal"></i>
32 <span ID:</span>
33 <span th:text="{curso.getCursoID()}"></span>
34 </h6>
35 <h6>
36 <i class="fa fa-clock"></i>
37 <span th:text="{curso.getCursoDuracion()}"></span>
38 </h6>
39 <h6>
40 <i class="fa fa-group"></i>
41 <span th:text="{curso.getEstudiante().size()}"></span>
42 <span>Estudiantes</span>
43 </h6>
44 <h6>
45 <i class="fas fa-fingerprint"></i>
46 <span th:text="{curso.getInstructor().getNombres()}" + ' ' + "{curso.getInstructor().getApellidos()}"></span>
47 </h6>
48 <p th:text="{curso.getCursoDescripcion()}"></p>
49 <div class="botones">
50 <div>
51 <a class="btn btn-danger" onclick="return confirm('El curso se Eliminara, Esta seguro')" th:hc
52 </div>
53 <div>
54 <a class="btn btn-success" th:href="@{/cursos/formUpdate(cursoId={curso.getCursoId()})}">Actu
55 </div>
56 </div>
57 </div>
58 </li>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>

```

El controlador finalmente es el encargado de responder a los eventos solicitados desde la vista y realizar en caso que sea necesario peticiones al modelo, aunque también este podría dar respuesta a la vista sin necesidad de hacer comunicación con el modelo.

Ilustración 9: Modulo controlador

```

CursoControlador.java X
1 package com.courseConnect.admin.web;
2
3 import java.io.IOException;
25
26 @Controller
27 @RequestMapping(value = "/cursos")
28 public class CursoControlador {
29
30     @Autowired
31     private CursoServicio cursoServicio;
32
33     @Autowired
34     private InstructorServicio instructorServicio;
35
36     @Autowired
37     private ContenidoDao contenidoDao;
38
39     @Autowired
40     private ContenidoServicio contenidoServicio;
41
42     public CursoControlador(CursoServicio cursoServicio) {
43         this.cursoServicio = cursoServicio;
44     }
45
46     @GetMapping(value = "/index")
47     public String cursos(Model model, @RequestParam(name = "keyword", defaultValue = "") String keyword) {
48         List<Curso> cursos = cursoServicio.encontrarCursosPorNombre(keyword);
49         model.addAttribute("listCursos", cursos);
50         model.addAttribute("keyword", keyword);
51         return "curso-views/cursos";
52     }
53
54     @GetMapping(value = "/delete")
55     public String deleteCursos(Long cursoId, String keyword) {
56         cursoServicio.removerCurso(cursoId);
57         return "redirect:/cursos/index?keyword=" + keyword;
58     }
59
60     @GetMapping(value = "/formUpdate")
61     public String updateCursos(Model model, Long cursoId) {
62         Curso curso = cursoServicio.cargarCursoPorId(cursoId);
63         Contenido contenido = new Contenido();
64         List<Instructor> instructors = instructorServicio.fetchInstructor();
65         model.addAttribute("curso", curso);
66         model.addAttribute("listInstructors", instructors);
67         model.addAttribute("contenido", contenido);
68         return "curso-views/formUpdate";
69     }
70
71     @GetMapping(value = "/formCrear")
72     public String formCrear(Model model) {
73         List<Instructor> instructors = instructorServicio.fetchInstructor();
74         model.addAttribute("listInstructors", instructors);
75         model.addAttribute("curso", new Curso());
76         model.addAttribute("contenido", new Contenido());
77         return "curso-views/formCrear";
78     }
--

```

6.5 Pruebas

JUnit 5 y Mockito

Son dos bibliotecas de pruebas muy populares en el mundo de la programación en Java. Se utilizan principalmente para escribir y ejecutar pruebas unitarias y pruebas de integración en aplicaciones (EduEsqui, 2023).

En este proyecto a nivel de pruebas está enfocado en Unitarias ya que son las más importantes y con la ayuda de la biblioteca de mockito ayuda a la automatización de las pruebas y aumentara la calidad del producto.

JUnit 5: Nos permite evaluar el resultado de la ejecución de un método. Es decir, nos permite comparar el resultado esperado, con el que realmente estamos obteniendo después de ejecutar el método

Ilustración 10: Pruebas unitarias I

```
@Test
public void cargarCursoPorId() {
    when(cursoDao.findById(any(Long.class))).thenReturn(Optional.of(curso));

    Curso respuesta = servicio.cargarCursoPorId(cursoId);
    assertNotNull(respuesta);
    assertEquals(1L, respuesta.getCursoId());

    verify(cursoDao, times(1)).findById(any(Long.class));
}
```

Mockito: Nos ayuda a simular la respuesta de otro método necesario para ejecutar el método que necesitamos probar. Estos nos permiten centrarnos únicamente en el método que deseamos probar.

Ilustración 11: Pruebas Unitarias II

```
@ExtendWith(MockitoExtension.class)
public class CursoServicioImplTest {

    @InjectMocks
    private CursoServicioImpl servicio;

    @Mock
    private CursoDao cursoDao;

    @Mock
    private InstructorDao instructorDao;

    @Mock
    private EstudianteDao estudianteDao;
```

7. Conclusiones

- El utilizar herramientas enfocadas en soluciones web y formularios, ayudo a tener visión mucho más clara de los objetivos que debe cumplir el proyecto, además ayudo organizar la planificación de los tiempos del proyecto.
- El desarrollo del software permitió adquirir conocimientos que permitieran en un futuro desarrollar herramientas más competitivas y prácticas que incrementen los resultados obtenidos actualmente.
- La aplicación esta desarrollada para ampliaciones que pudieran a futuro implementarse sobre ella.

8. Recomendaciones

- Se recomienda hacer una revisión periódica sobre las tecnologías usadas y hacer su ajuste de versiones para evitar a futuro conflictos o incompatibilidad de versiones.
- Es importante en el momento de realizar el desarrollo de algún software, tener bien definidos los procesos lógicos y la descripción detallada de los requerimientos, para no generar atrasos en el momento de la implementación.
- Se recomienda que se implementa una caja de comentarios de los cursos para que el usuario pueda comentar de ellos.
- Se recomienda que se implemente módulos de videos personalizados sobre el curso por el instructor hacia los estudiantes.
- Se recomienda continuar con este proyecto de grado ya que permite que los estudiantes profundicen sus conocimientos y fomenten su investigación.

9. Referencias

Bootstrap. (5 de Agosto de 2023). Obtenido de <https://getbootstrap.com/>
Fernández. (2018). *Using Thymeleaf*.

GCF Global. (2023). *¿Qué es la educación virtual?* Obtenido de Educación virtual: (Economista, 2020)<https://edu.gcfglobal.org/es/educacion-virtual/que-es-la-educacion-virtual/1/#>
Gomez. (2000). *Introducción a HTML y JavaScript*. Alfaomega.

Quesenbery, S. H. (2013). *A Web for Everyone: Designing Accessible User Experiences*.
Walls, C. (2021). *Spring in Action*.

Wikipedia. (1 de Septiembre de 2023). Obtenido de
[https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

Republica, L. (03 de 03 de 2023). *www.larepublica.co*. Obtenido de <https://www.larepublica.co/alta-gerencia/networking-y-desarrollo-web-la-nueva-oferta-de-cursos-cortos-de-las-universidades-3559762#:~:text=Seg%C3%BAun%20estudio%20de%20la,la%20cifra%20ascendi%C3%B3%20a%2070%25>.

EduEsqui. (06 de 10 de 2023). *FuncionaEnMiMaquina*. Obtenido de
<https://funcionaenmimaquina.com/aprende-a-crear-pruebas-unitarias-con-junit-y-mockito-en-15-minutos/>

Salas, I. (31 de 10 de 2008). <https://observatoriotecedu.uned.ac.cr/>. Obtenido de
<https://observatoriotecedu.uned.ac.cr/webct/>

SánchezRodríguez, J. (01 de 2009). PLATAFORMAS DE ENSEÑANZA VIRTUAL PARA ENTORNOS EDUCATIVOS.

10. Anexos

Anexo A. Manual del usuario