**Univermilenium**

Ingeniería En Sistemas Computacionales

Plantel: Rectoría

Ingeniería De Software

Profesor: Alejandro Vargas Romero

Actividad final.

Luis Enrique Piedra Casiano

GRUPO: E1LISC801

Febrero 2024

# Introducción.

En este proyecto en el que se desarrollará una página web destinada a ser un repositorio universitario específicamente para los alumnos de la universidad Univermilenium. En este espacio digital, nos dedicaremos en cumplir la misión de centralizar y organizar de manera simple y accesible los recursos académicos, investigaciones y trabajos de los estudiantes.

Nuestra plataforma web busca no solo consolidar la colaboración y el intercambio de ideas de la comunidad universitaria, sino también, la facilitación de la elaboración de trabajos, tesis, guías, etc. Por medio de trabajos anteriores, de otros estudiantes, guías validadas, documentos y artículos aprobados por la comunidad universitaria. A través de una interfaz intuitiva, categorías por las cual podrás navegar, descubrir proyectos destacados según tu licenciatura y acceder a valiosos recursos que enriquecerán su experiencia académica.

En este repositorio académica destacaran la diversidad de disciplinas y enfoques académicos. Desde tesis innovadoras hasta proyectos colaborativos, cada contribución representa un pilar fundamental de nuestro compromiso con la excelencia educativa. Además, ofrecemos herramientas de búsqueda avanzada y funciones interactivas para facilitar la exploración y la conexión entre estudiantes, profesores e investigadores. Únete a nosotros en este viaje digital, donde el aprendizaje se convierte en una experiencia dinámica y colaborativa que trasciende las barreras físicas del campus universitario.

# Definiciones y requerimientos

## Definición general del proyecto

Nuestro proyecto consiste en la creación de un repositorio universitario en una pagina web, es decir, una plataforma digital que agrupa y organiza todo tipo de conocimiento generado por la comunidad de la universidad. En este espacio no solo servirá como archivo central, sino que también promueve la colaboración, intercambio de ideas y la conectividad entre estudiantes, profesores, investigadores o administrativos.

Buscamos establecer un ambiente virtual que facilite el acceso a recursos académicos, fomente la investigación y contribuya al enriquecimiento continuo de la experiencia educativa.

En este repositorio digital se constituye como un punto de encuentro electrónico donde se juntan diversas disciplinas y áreas de estudio. Desde trabajos de investigación y proyectos académicos hasta material didáctico, cada contribución representa un eslabón valioso en la cadena del conocimiento.

Además, nuestra plataforma no solo aspira a ser un depósito estático, sino un espacio dinámico donde la comunidad pueda interactuar, comentar y colaborar en tiempo real. A través de una interfaz amigable y herramientas de búsqueda, pretendemos facilitar el descubrimiento y la difusión del conocimiento de cada una de las licenciaturas del plantel, convirtiendo esta página web en un recurso vital para el crecimiento intelectual de nuestra institución.

## Objetivo general

Establecer un repositorio institucional en una pagina web que sirva como un hub integral para la gestión, acceso y difusión del conocimiento académico generado por los integrantes de la comunidad universitaria.

Buscamos crear un entorno colaborativo y dinámico que promueva la interacción entre estudiantes, profesores e investigadores, facilitando el intercambio de ideas, la exploración de proyectos y la consolidación de recursos educativos. Con un enfoque en la accesibilidad y la innovación, aspiramos a potenciar el aprendizaje continuo y a fortalecer la identidad académica de nuestra institución.

## Objetivos específicos

1. **Centralización de Recursos**: Juntar y organizar de manera sistemática y accesible los trabajos de investigación, tesis, proyectos académicos y material didáctico generados por la comunidad universitaria.
2. **Fomentar la Colaboración Académica**: Proporcionar un espacio interactivos y funciones que faciliten la interacción entre estudiantes, profesores e investigadores, promoviendo la discusión y la colaboración en tiempo real.
3. **Incentivar la Contribución Continua**: Establecer mecanismos que motiven a la comunidad a contribuir activamente al repositorio, ya sea mediante la subida de nuevos contenidos, revisiones o actualizaciones, creando así una base de conocimiento en constante evolución.
4. **Promover la Difusión del Conocimiento**: Implementar estrategias de difusión que aumenten la visibilidad del repositorio tanto a nivel interno como externo, posicionándolo como un recurso valioso para la comunidad académica y la sociedad en general.
5. **Mantenimiento y Actualización:** Garantizar la sostenibilidad y relevancia a largo plazo del repositorio mediante un plan de mantenimiento regular, actualizando tecnologías, corrigiendo errores y asegurando la integridad del contenido.
6. **Medir Impacto y Retroalimentación:** Implementar métricas de seguimiento para evaluar la eficacia y el impacto del repositorio, y recopilar retroalimentación de los usuarios para realizar mejoras continuas en la plataforma.

# Procedimientos y requerimientos

## Software

### HTML

El lenguaje HTML (Hypertext Markup Language) desempeña un papel fundamental en nuestro proyecto como el lenguaje de marcado esencial para la creación y estructuración de páginas web.

Utilizamos HTML para definir la organización y presentación de la información en nuestro repositorio universitario. A través de etiquetas y elementos, creamos una jerarquía clara que permite la inclusión de texto, enlaces, imágenes y otros recursos multimedia, facilitando la creación de una interfaz amigable e interactiva. La semántica de HTML también contribuye a mejorar la accesibilidad y la indexación del contenido, asegurando que nuestro repositorio sea eficiente, comprensible y compatible con diversas plataformas y dispositivos.

Según el blog de desarrolladoresweb (*HTML*, s. f.) define el lenguaje HTML como:

HTML es un lenguaje de marcación que sirve para definir el contenido de las páginas web. Se compone en base a etiquetas, también llamadas marcas o tags, con las cuales conseguimos expresar las partes de un documento, cabecera, cuerpo, encabezados, párrafos, etc. En definitiva, el contenido de una página web. (párr. 2)

### CSS

CSS, o Cascading Style Sheets, desempeña un papel esencial en nuestro proyecto al proporcionar un medio para diseñar y estilizar la presentación de nuestras páginas web. Utilizando reglas de estilo, CSS define cómo se deben visualizar los elementos HTML, controlando aspectos como el diseño, color, fuente y espaciado.

Esto permite una separación efectiva entre la estructura del contenido (HTML) y su presentación (CSS), facilitando la coherencia visual y la adaptabilidad de nuestro repositorio universitario a través de diferentes dispositivos y tamaños de pantalla.

## La capacidad de cascada en CSS permite establecer prioridades y heredar estilos, proporcionando flexibilidad y eficiencia en el diseño de la interfaz.

Según el blog lenguajescss (*¿Qué es CSS? - CSS en español*, s. f.) define el lenguaje css como:

Un «idioma» o lenguaje, que nos permite aplicar estilos (colores, formas, márgenes, etc...) a uno o varios documentos (generalmente documentos HTML, páginas webs) de forma automática y masiva.

Se le denomina estilos en cascada porque se lee, procesa y aplica el código desde arriba hacia abajo (siguiendo un patrón denominado herencia que trataremos más adelante) y en el caso de existir ambigüedad (código que se contradice), se siguen una serie de normas para resolver dicha ambigüedad. (párr. 5-6).

### JavaScript

JavaScript, un componente fundamental en nuestro proyecto, es el lenguaje de programación versátil y dinámico diseñado para proporcionar interactividad y funcionalidad a las páginas web.

Utilizamos JavaScript para mejorar la experiencia del usuario en nuestro repositorio universitario, permitiendo la manipulación del contenido en tiempo real, la validación de formularios, y la creación de interfaces interactivas.

Además, aprovechamos su capacidad para comunicarse con el servidor, facilitando la carga dinámica de datos y mejorando la eficiencia de la plataforma.

Con JavaScript, buscamos crear una experiencia de usuario fluida e intuitiva, elevando la usabilidad y la interconexión dentro de nuestra comunidad académica en línea.

La página AWS de Amazon (*¿Qué es JavaScript? - Explicación de JavaScript (JS) - AWS*, s. f.) define el lenguaje JavaScript como:

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web. Por ejemplo, al navegar por Internet, en cualquier momento en el que vea un carrusel de imágenes, un menú desplegable “click-to-show” (clic para mostrar), o cambien de manera dinámica los elementos de color en una página web, estará viendo los efectos de JavaScript. (párr. 1)

### XAMPP

XAMPP es una herramienta esencial en nuestro proyecto, ya que nos proporciona un entorno de herramientas de desarrollo local completo que facilita la creación y prueba de nuestra página web.

XAMPP, tiene distintas herramientas o entorno en el cual se divide, el cual sumismo nombre describe, "X" (cualquier sistema operativo), "Apache" (servidor web), "MySQL" (base de datos), "PHP" (lenguaje de scripting) y "Perl" (lenguaje de programación), integra estos componentes esenciales para el desarrollo web en un paquete fácil de instalar y de activar.

Este recurso nos permite simular un servidor web en nuestra máquina local, facilitando la implementación y prueba de funcionalidades antes de desplegar la página en un entorno de producción.

La inclusión de Apache como servidor web, MySQL para la gestión de bases de datos y PHP para la ejecución de scripts del lado del servidor garantiza un entorno de desarrollo robusto y versátil, fundamental para la creación exitosa de nuestro repositorio universitario en línea.

Según Jesús (2022) en el blog Dongee define la herramienta XAMPP como:

Xampp es un servidor web local multiplataforma que permite la creación y prueba de páginas web u otros elementos de programación. Sin embargo, Xampp integra una serie de herramientas que potencian y facilitan la experiencia al desarrollador. Es decir, Xampp en sí mismo no es un programa, sino un paquete de programas o software que contiene herramientas de gestión de base de datos.

Desarrollado por Apache Friends y su nombre es un acrónimo de cada una de las principales herramientas que contiene; esto nos sirve para hacernos una idea de cuáles son los principales programas que vienen en el paquete. (párr. 4-6).

### Visual Studio Code

Visual Studio Code es una herramienta crucial en nuestro proyecto, sirviendo como un entorno de desarrollo integrado (IDE) altamente versátil.

Este editor de código fuente, desarrollado por Microsoft, ofrece una interfaz intuitiva y características potentes que mejoran la eficiencia y la calidad del desarrollo.

Con Visual Studio Code, los desarrolladores tienen acceso a funciones como resaltado de sintaxis, completado automático, depuración integrada y control de versiones, facilitando la escritura y el mantenimiento del código. Su soporte extensivo para una variedad de lenguajes de programación y su capacidad para integrarse con herramientas y extensiones adicionales lo convierten en una elección ideal para el desarrollo de nuestro repositorio universitario.

Además, su naturaleza de código abierto y su activa comunidad de usuarios aseguran actualizaciones frecuentes y un entorno de desarrollo moderno.

Según la página oficial de Microsoft (Visual Studio: IDE y editor de código para desarrolladores de software y teams, 2023), define el IDE como:

Visual Studio Code es un editor de código fuente ligero pero eficaz que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux.

Incluye compatibilidad integrada con JavaScript, TypeScript y Node.js, y cuenta con un amplio ecosistema de extensiones para otros lenguajes y entorno de ejecución (como C++, C#, Java, Python, Go, .NET). (párr. 7)

## Hardware

### IdeaPad 3 6ta Gen (15.6", AMD)

Para el desarrollo de este proyecto fue esencial un equipo de cómputo, proporcionando la plataforma principal para el desarrollo, prueba y gestión de nuestro repositorio universitario en línea. Este dispositivo portátil ofrece la flexibilidad y la movilidad necesarias para trabajar en diferentes entornos.

Con la capacidad de ejecutar software de desarrollo, editores de código y herramientas de gestión de proyectos, la laptop se convierte en el centro de operaciones para el equipo de desarrollo. Su pantalla, teclado y touchpad contribuyen a la comodidad y productividad del usuario. Además, la conectividad a Internet facilita la colaboración en línea y la integración de recursos externos en nuestro proyecto.

La laptop, al ser una herramienta versátil y portátil, permite a los miembros del equipo trabajar eficientemente desde diversos lugares, fomentando la colaboración y la continuidad del desarrollo del repositorio universitario.

Sus especificaciones:

Edición Windows 11 Home Single Language

Versión 22H2

Procesador AMD Ryzen 7 5700U with Radeon Graphics 1.80 GHz

RAM instalada 16.0 GB (13.8 GB usable)

Tipo de sistema Sistema operativo de 64 bits, procesador basado en x64

**Arquitectura del Sistema**

* descripción de jerarquías

En el repositorio univer, la jerarquía se organiza en términos de repositorios, ramas, commits y archivos. Puedes representar esta jerarquía utilizando un modelo relacional. Aquí hay un ejemplo simplificado de cómo podrías estructurar un modelo relacional para GitHub:

Tabla de Usuarios:

UsuarioID (Clave primaria)

NombreUsuario

CorreoElectronico

Tabla de Repositorios:

RepoID (Clave primaria)

NombreRepositorio

Descripcion

UsuarioID (Clave foránea referenciando a la tabla de Usuarios)

Tabla de Ramas:

RamaID (Clave primaria)

NombreRama

RepoID (Clave foránea referenciando a la tabla de Repositorios)

Tabla de Commits:

CommitID (Clave primaria)

Mensaje

Fecha

RamaID (Clave foránea referenciando a la tabla de Ramas)

Tabla de Archivos:

ArchivoID (Clave primaria)

NombreArchivo

RutaArchivo

CommitID (Clave foránea referenciando a la tabla de Commits)

**En este modelo:**

Un usuario puede tener varios repositorios (relación uno a muchos entre Usuarios y Repositorios).

Un repositorio puede tener varias ramas (relación uno a muchos entre Repositorios y Ramas).

Una rama puede tener varios commits (relación uno a muchos entre Ramas y Commits).

Un commit puede afectar varios archivos (relación uno a muchos entre Commits y Archivos).

Esta es solo una representación básica y simplificada. En un entorno de desarrollo real, es posible que necesites considerar otras entidades y relaciones, como colaboradores en un repositorio, problemas (issues), solicitudes de extracción (pull requests), entre otros. Además, las relaciones podrían ser más complejas dependiendo de la estructura y las características específicas de tu aplicación.

**Modulos**

Diseñar módulos basados en ingeniería de software implica organizar y estructurar el sistema de manera que cada módulo tenga una responsabilidad clara y esté lo más desacoplado posible de otros módulos. Aquí hay una propuesta de cómo podrías organizar los módulos para el sistema de GitHub que describiste:

**Módulo de Usuarios:**

Clases: Usuario

Responsabilidades:

Gestionar la información de los usuarios.

Autenticación y autorización.

Interactuar con la base de datos para almacenar y recuperar información de usuarios.

**Módulo de Repositorios:**

Clases: Repositorio

Responsabilidades:

Gestionar la información de los repositorios.

Manejar la relación entre usuarios y repositorios.

Interactuar con la base de datos para almacenar y recuperar información de repositorios.

**Módulo de Ramas:**

Clases: Rama

Responsabilidades:

Gestionar la información de las ramas.

Manejar la relación entre repositorios y ramas.

Interactuar con la base de datos para almacenar y recuperar información de ramas.

**Módulo de Commits:**

Clases: Commit

Responsabilidades:

Gestionar la información de los commits.

Manejar la relación entre ramas y commits.

Interactuar con la base de datos para almacenar y recuperar información de commits.

**Módulo de Archivos:**

Clases: Archivo

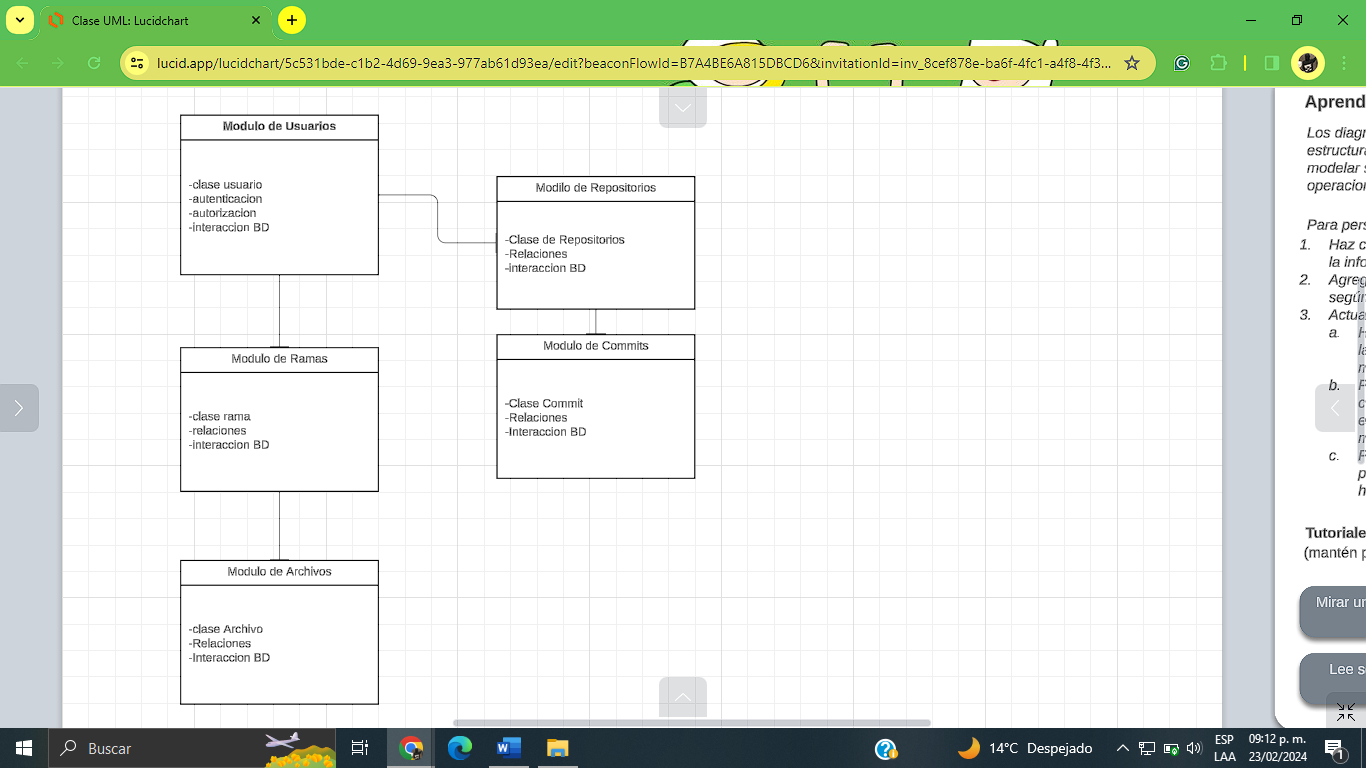
Responsabilidades:

Gestionar la información de los archivos.

Manejar la relación entre commits y archivos.

Interactuar con la base de datos para almacenar y recuperar información de archivos.

Cada módulo debe encapsular su funcionalidad y exponer interfaces bien definidas para la interacción con otros módulos. Además, puedes considerar la implementación de patrones de diseño como el patrón Repositorio para abstraer la capa de persistencia y facilitar cambios futuros en la base de datos.



### Referencias

*HTML*. (s. f.). Desarrollo Web. <https://desarrolloweb.com/home/html>

*¿Qué es CSS? - CSS en español*. (s. f.). Lenguaje CSS. <https://lenguajecss.com/css/introduccion/que-es-css/>

*¿Qué es JavaScript? - Explicación de JavaScript (JS) - AWS*. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/javascript/>

Jesús. (2022, 18 noviembre). *Conoce qué es Xampp y por qué deberías usarlo en tus proyectos*. Tutoriales Dongee. <https://www.dongee.com/tutoriales/que-es-xampp/>