

Universidad de San Carlos de Guatemala.

Centro Universitario de Occidente.

División de Ciencias de la Ingeniería.

Teoría de Sistemas I.

Ing. Pedro Domingo.



“DOCUMENTACIÓN TÉCNICA”

Diego José Maldonado Monterroso.

Carné: 201931811.

Quetzaltenango, Guatemala.

23 de agosto de 2023.

INTRODUCCIÓN

En el vasto mundo del desarrollo web, se encuentran diversas tecnologías y herramientas que desempeñan un papel fundamental en la creación y el funcionamiento de sitios web modernos y dinámicos. Estas tecnologías y herramientas forman la columna vertebral de la mayoría de los proyectos web y son esenciales para desarrolladores, diseñadores y administradores de sistemas por igual.

En este trabajo exploraremos una de las tantas formas de desarrollo, así como un ejemplo de como podemos plantear nuestros proyectos antes de plasmar nuestras ideas como código.

MARCO TEÓRICO

Capítulo 1.

1.1 Servidor HTTP Apache.

Apache HTTP Server, comúnmente conocido como Apache, es un servidor web de código abierto ampliamente utilizado en Internet. Es un software que recibe solicitudes de clientes web, como navegadores, y les envía las páginas web correspondientes.

Apache es una de las opciones más populares para alojar sitios web debido a su estabilidad, flexibilidad y capacidad de personalización. Es compatible con una amplia variedad de sistemas operativos y se integra bien con otros componentes de la pila LAMP (Linux, Apache, MySQL, PHP), que es esencial en el desarrollo web.

1.2 MySQL.

MySQL es un sistema de gestión de bases de datos (DBMS) de código abierto que permite almacenar, organizar y recuperar datos. Se utiliza principalmente para aplicaciones web y es compatible con múltiples lenguajes de programación.

MySQL es esencial para almacenar y gestionar datos en aplicaciones web. Se utiliza en combinación con PHP y un sinnúmero de lenguajes de programación para crear aplicaciones dinámicas que interactúan con bases de datos. Usa el lenguaje SQL diseñado para gestionar y manipular bases de datos relacionales. Fue desarrollado en la década de 1970 y se ha convertido en un estándar en la gestión de bases de datos. SQL se utiliza para realizar una variedad de tareas relacionadas con datos, como la creación, modificación y consulta de bases de datos.

1.3 PHP.

PHP (Hypertext Preprocessor) es un lenguaje de programación de código abierto diseñado para el desarrollo web. Se utiliza para crear aplicaciones web dinámicas y se puede integrar fácilmente con HTML.

PHP es un lenguaje esencial para la creación de sitios web dinámicos, ya que permite a los desarrolladores interactuar con bases de datos, procesar formularios y generar contenido web en tiempo real.

1.4 HTML.

HTML (Hypertext Markup Language) es el lenguaje de marcado utilizado para crear páginas web. Define la estructura y el contenido de una página web mediante etiquetas.

HTML es el lenguaje base de la web y es esencial para la creación de páginas web. Define la estructura de una página y cómo se muestran los elementos en un navegador.

1.5 JavaScript.

JavaScript es un lenguaje de programación utilizado para agregar interactividad y funcionalidad a las páginas web. Se ejecuta en el navegador del usuario.

JavaScript es fundamental para crear sitios web interactivos y dinámicos. Permite realizar validaciones de formularios, animaciones, efectos visuales y más.

1.6 CSS.

CSS (Cascading Style Sheets) es un lenguaje utilizado para definir el estilo y la presentación de una página web. Permite controlar la apariencia de los elementos HTML.

CSS es esencial para el diseño web, ya que permite dar formato y estilo a los elementos HTML, lo que mejora la apariencia y la usabilidad de un sitio web.

1.7 Xampp.

XAMPP es un paquete de software que incluye Apache, MySQL, PHP y Perl, diseñado para facilitar la configuración de un entorno de desarrollo web local en sistemas Windows, macOS y Linux.

XAMPP es una herramienta valiosa para desarrolladores web, ya que permite crear y probar sitios web en un entorno de desarrollo local antes de desplegarlos en un servidor en vivo.

1.8 PhpMyAdmin.

PhpMyAdmin es una herramienta de administración de bases de datos web que proporciona una interfaz gráfica para administrar bases de datos MySQL a través de un navegador web.

PhpMyAdmin facilita la gestión y administración de bases de datos MySQL, lo que es crucial para los desarrolladores web que trabajan con aplicaciones que utilizan MySQL como motor de base de datos.

1.9 Visual Studio Code.

Visual Studio Code (VS Code) es un editor de código fuente de código abierto desarrollado por Microsoft. Es ampliamente utilizado en el desarrollo web y ofrece numerosas extensiones para admitir diferentes lenguajes de programación y herramientas.

VS Code es un entorno de desarrollo altamente personalizable y eficiente que facilita la escritura, depuración y gestión de código para proyectos web y aplicaciones.

Capítulo 2.

2.2 Backend.

Es la parte "oculta" de una aplicación web que se encuentra en el servidor. Realiza tareas que no son visibles para el usuario, como procesar solicitudes, administrar bases de datos, autenticar usuarios y gestionar la lógica empresarial. Los elementos clave del Backend son:

- **Servidor Web:** El servidor web, como Apache o Nginx, maneja las solicitudes entrantes desde el frontend y envía respuestas adecuadas, como páginas HTML, datos JSON o imágenes.
- **Lenguajes de Programación del Backend:** Los lenguajes de programación comunes para el backend incluyen PHP, Python, Ruby, Node.js (JavaScript en el lado del servidor) y más. Estos lenguajes se utilizan para crear la lógica de la aplicación y gestionar las operaciones del servidor.

- **Base de Datos:** El backend suele interactuar con una base de datos para almacenar y recuperar datos. Los sistemas de gestión de bases de datos, como MySQL, PostgreSQL o MongoDB, son comunes en el backend.
- **Autenticación y Seguridad:** El backend se encarga de la autenticación de usuarios, la autorización y la seguridad, asegurando que los datos y las operaciones estén protegidos.
- **API (Interfaz de Programación de Aplicaciones):** El backend a menudo expone API que permiten a aplicaciones frontend o terceros acceder a datos y funcionalidades de la aplicación de forma controlada.

En resumen, el Backend se ocupa de las tareas detrás de escena que hacen que una aplicación web funcione de manera efectiva. Ambos componentes son esenciales y trabajan juntos para proporcionar una experiencia completa en una aplicación web.

2.3 Frontend.

Es la parte de una aplicación web que los usuarios ven y con la que interactúan directamente en sus navegadores web. Incluye la interfaz de usuario, el diseño, la presentación y la experiencia del usuario. Los elementos clave del frontend son:

- **HTML (Hypertext Markup Language):** HTML es el lenguaje de marcado utilizado para estructurar y crear la parte visual de una página web. Define la disposición de los elementos en una página, como encabezados, párrafos, imágenes, formularios, enlaces y más.

- CSS (Cascading Style Sheets): CSS se utiliza para dar estilo y formato a los elementos HTML. Define la apariencia de la página, incluidos aspectos como colores, fuentes, márgenes, espaciado y diseño general.
- JavaScript: JavaScript es un lenguaje de programación que agrega interactividad y funcionalidad dinámica a una página web. Permite crear efectos visuales, realizar validaciones de formularios, cargar datos de forma asincrónica y mucho más.
- Frameworks y Bibliotecas: Los desarrolladores Frontend a menudo utilizan Frameworks y bibliotecas como React, Angular o Vue.js para simplificar y agilizar el desarrollo de aplicaciones web interactivas y ricas en contenido.

El Frontend es lo que los usuarios ven y con lo que interactúan directamente al visitar un sitio web. Su objetivo principal es proporcionar una experiencia de usuario atractiva y funcional.

Capítulo 3.

3.1 Modelo Vista-Controlador (MVC).

El Modelo-Vista-Controlador (MVC) es un patrón de diseño arquitectónico ampliamente utilizado en el desarrollo web y de aplicaciones para separar de manera eficiente las diferentes responsabilidades y componentes de una aplicación. Este patrón ayuda a organizar el código de una manera que mejora la modularidad, la reutilización y la mantenibilidad de la aplicación. A continuación, se define el MVC en el contexto del desarrollo web:

1. Modelo (Model).

El Modelo representa la capa de datos y la lógica de negocio de la aplicación. En esta capa, se gestionan los datos, su almacenamiento, recuperación y procesamiento.

Los modelos suelen comunicarse con la base de datos o cualquier otro almacenamiento de datos para obtener o actualizar información.

No deben tener conocimiento de la interfaz de usuario ni de la lógica de presentación. Son independientes de cómo se muestran los datos.

2. Vista (View).

La Vista es la capa encargada de la presentación de los datos al usuario. Define cómo se muestran los datos al cliente.

Las Vistas son pasivas y reflejan el estado actual del Modelo. No realizan operaciones de lógica de negocio ni manipulan directamente los datos.

Pueden existir múltiples vistas para un mismo modelo, lo que permite mostrar la misma información de diferentes maneras.

3. Controlador (Controller):

El Controlador actúa como intermediario entre el Modelo y la Vista. Recibe las solicitudes del usuario a través de la interfaz de usuario y gestiona las acciones correspondientes.

Se encarga de la lógica de negocio, toma decisiones basadas en la entrada del usuario y actualiza el Modelo en consecuencia.

También actualiza la Vista para reflejar los cambios en el Modelo, lo que permite que los datos se presenten de manera coherente y actualizada.

En resumen, el patrón MVC divide una aplicación web en tres componentes principales, cada uno con una responsabilidad claramente definida:

El Modelo se encarga de la gestión de datos y la lógica de negocio.

La Vista se encarga de la presentación y la interfaz de usuario.

El Controlador se encarga de coordinar las acciones del usuario, la lógica de negocio y la actualización de la Vista y el Modelo.

Esta separación de preocupaciones facilita el desarrollo colaborativo, la escalabilidad y la mantenibilidad de las aplicaciones web, ya que cada componente puede ser desarrollado, probado y modificado de forma independiente.

OBJETIVOS

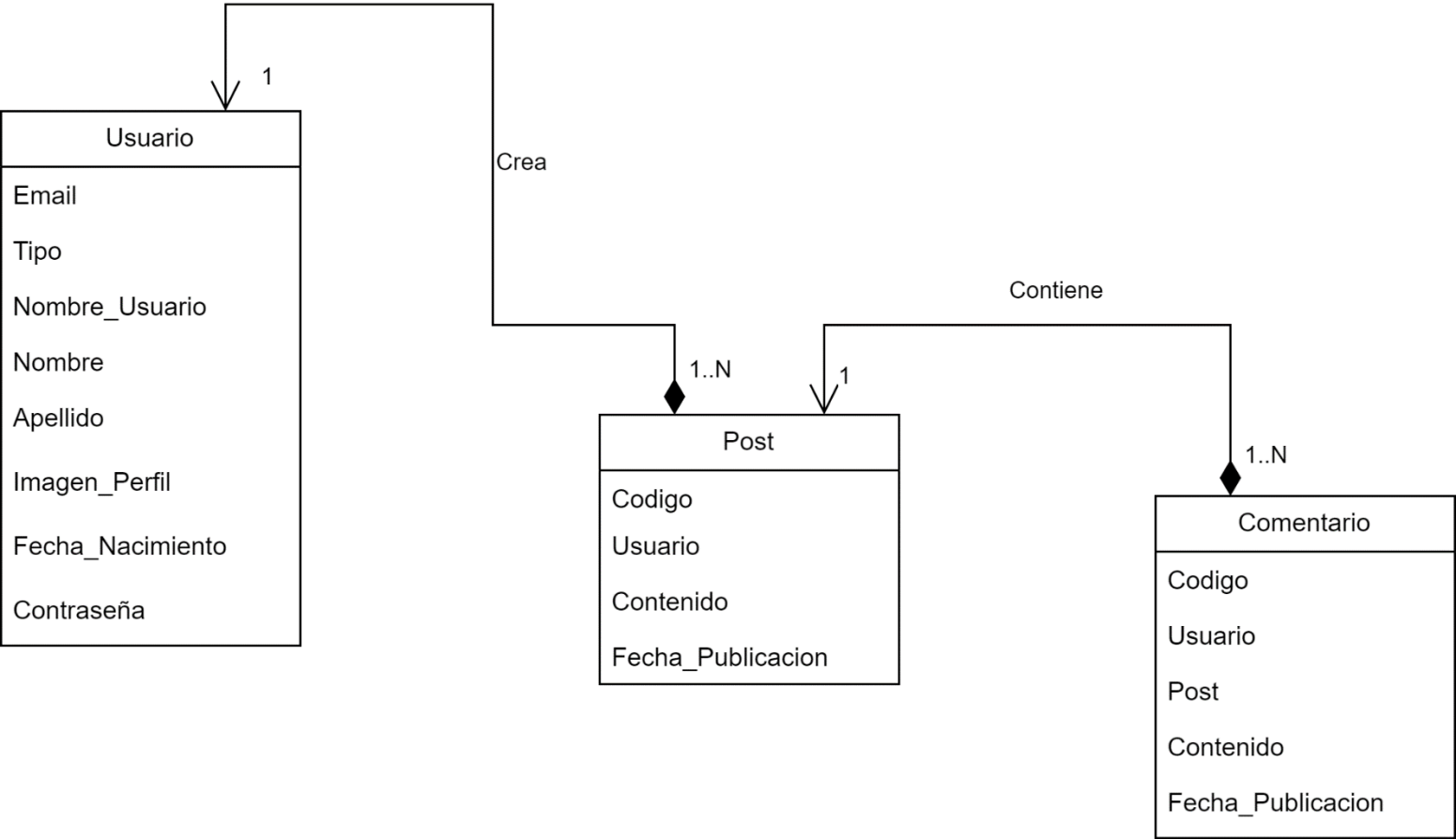
Objetivo General.

- Explorar el mundo del desarrollo web con las tecnologías más comunes y sin el uso de algún framework.

Objetivo Especifico:

- Realizar un Blog, donde los clientes puedan hacer posts e interactuar con comentarios, usando JavaScript, HTML y CSS para el área de Frontend, así como PHP y MySQL para el área de Backend.

MODELO DE DOMINIO



MODELO ENTIDAD RELACIÓN

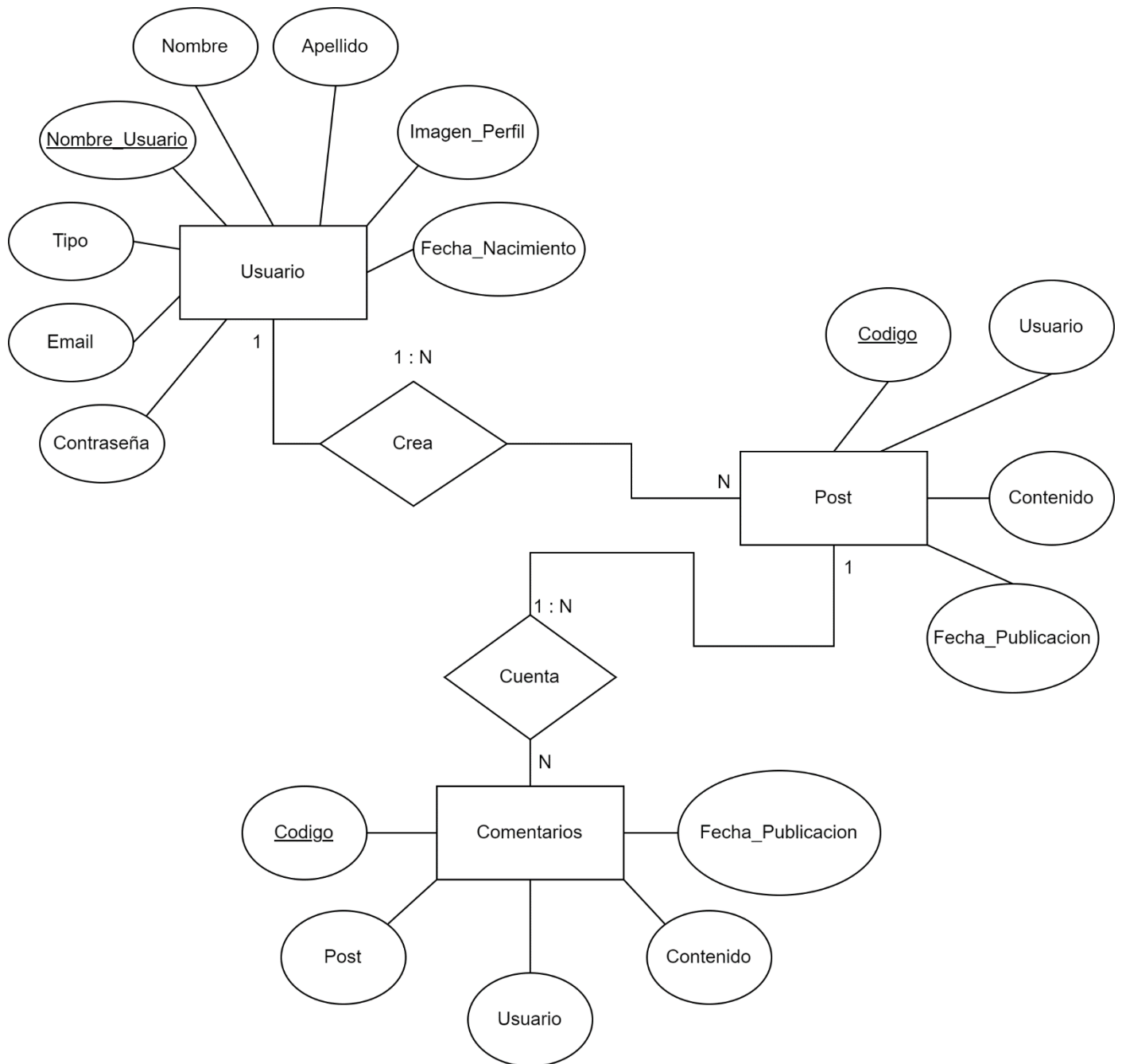
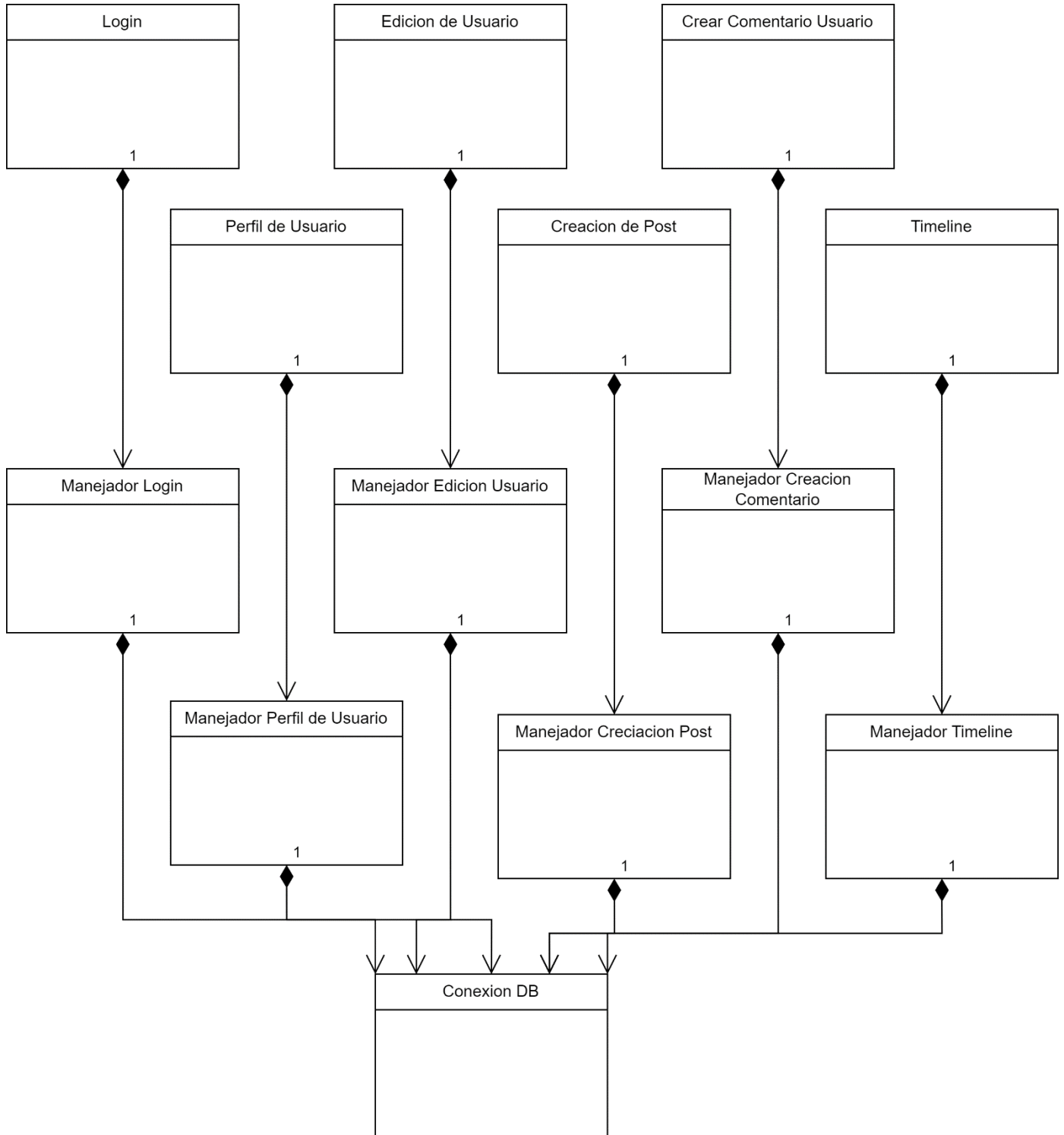
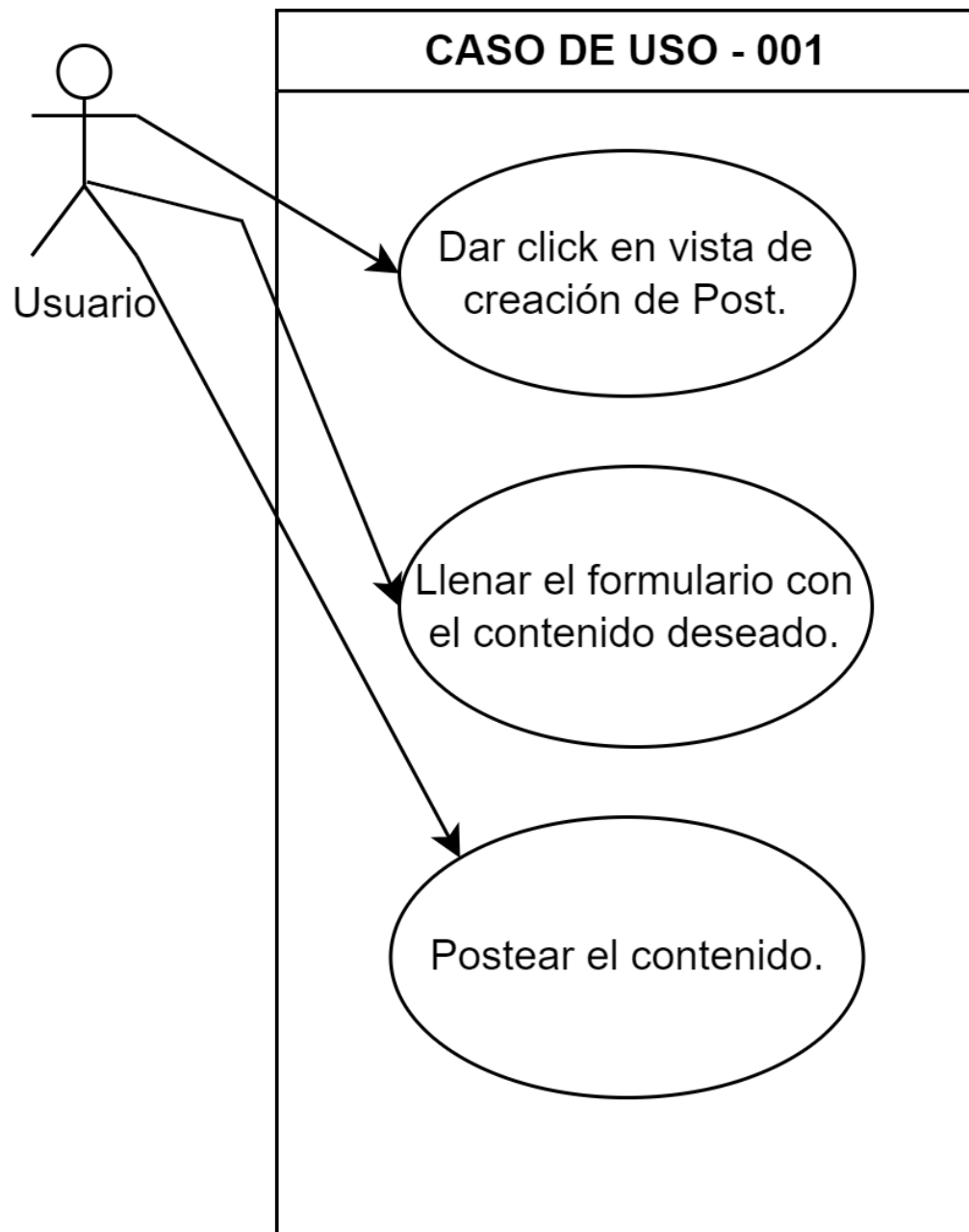
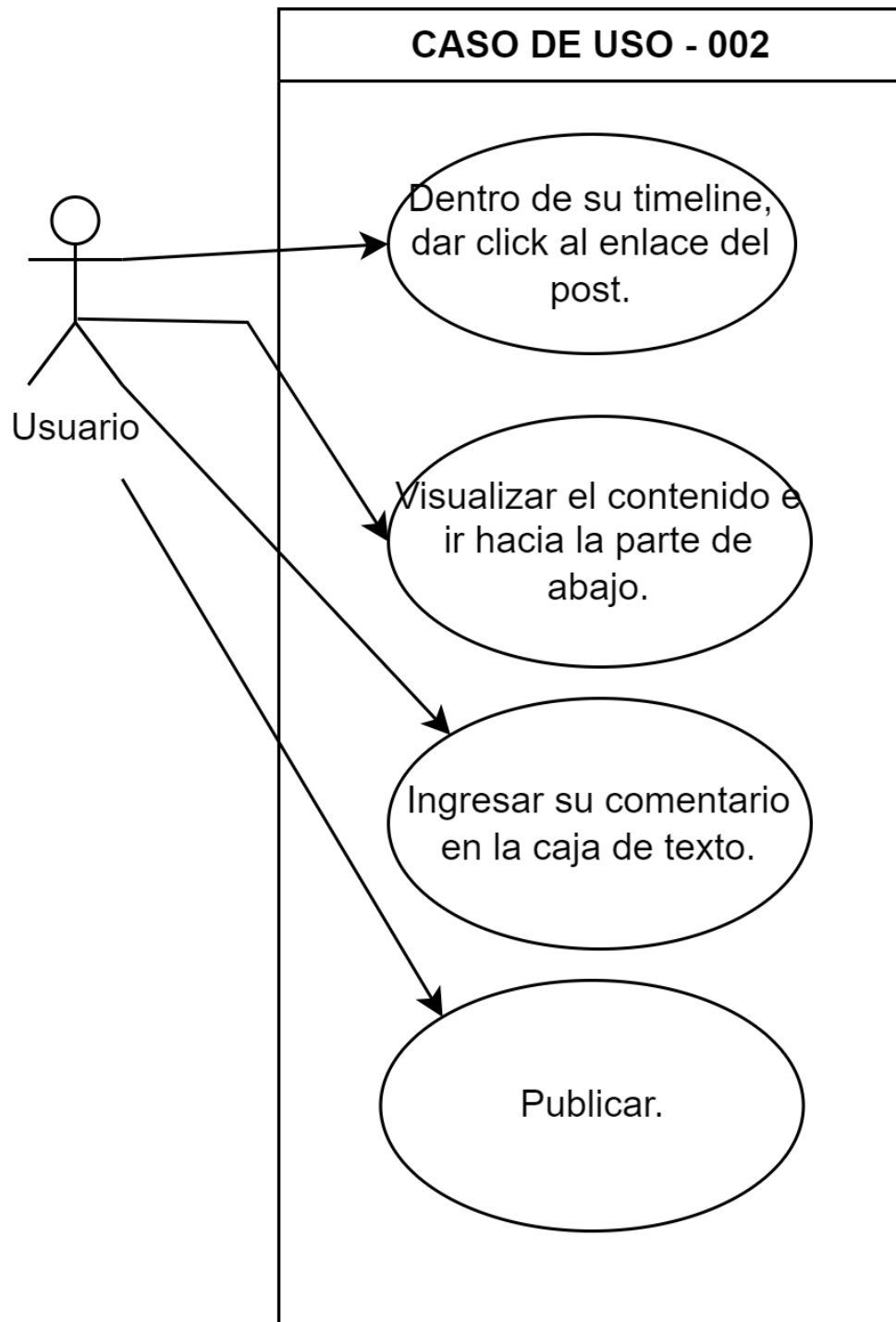


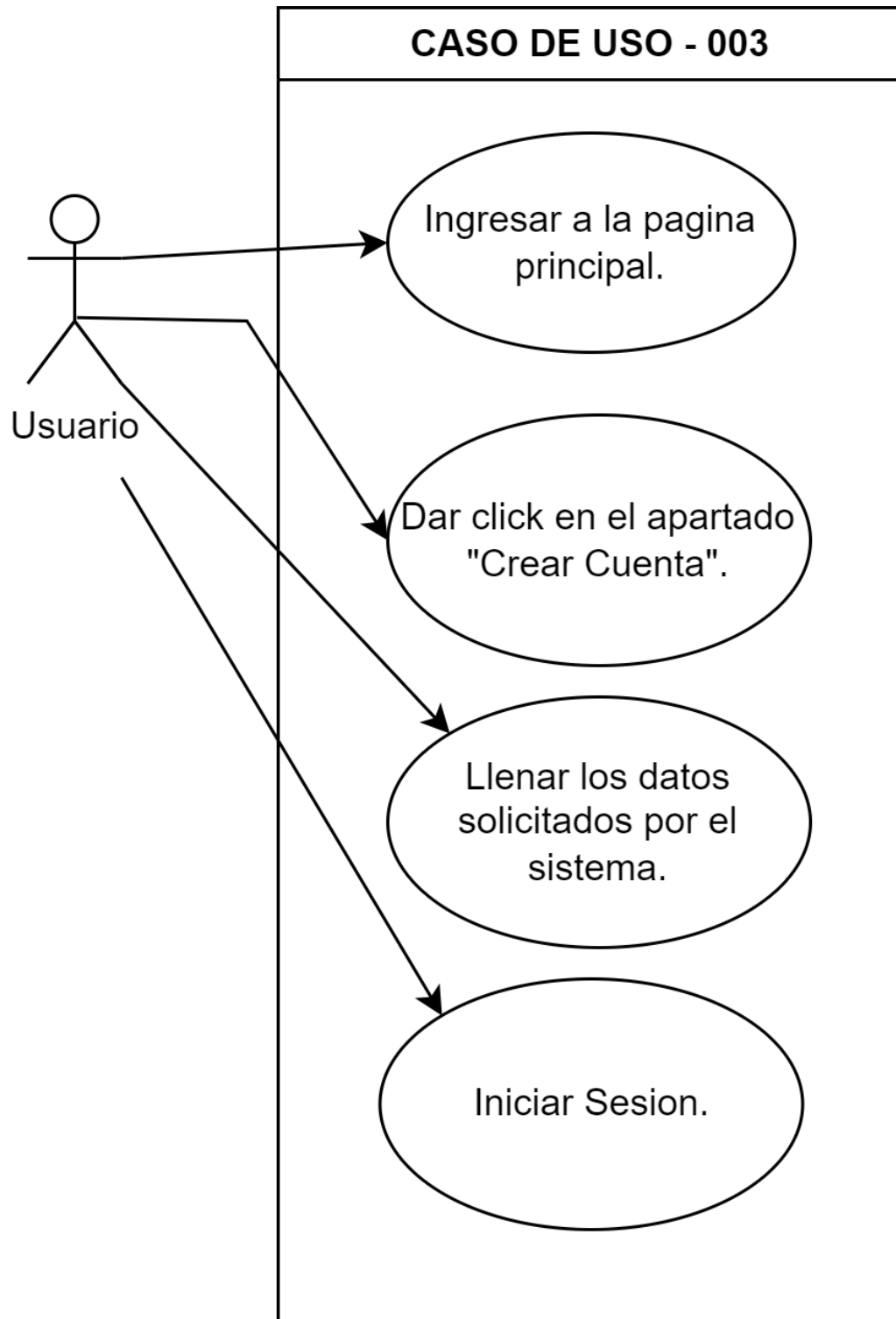
DIAGRAMA DE CLASES UML

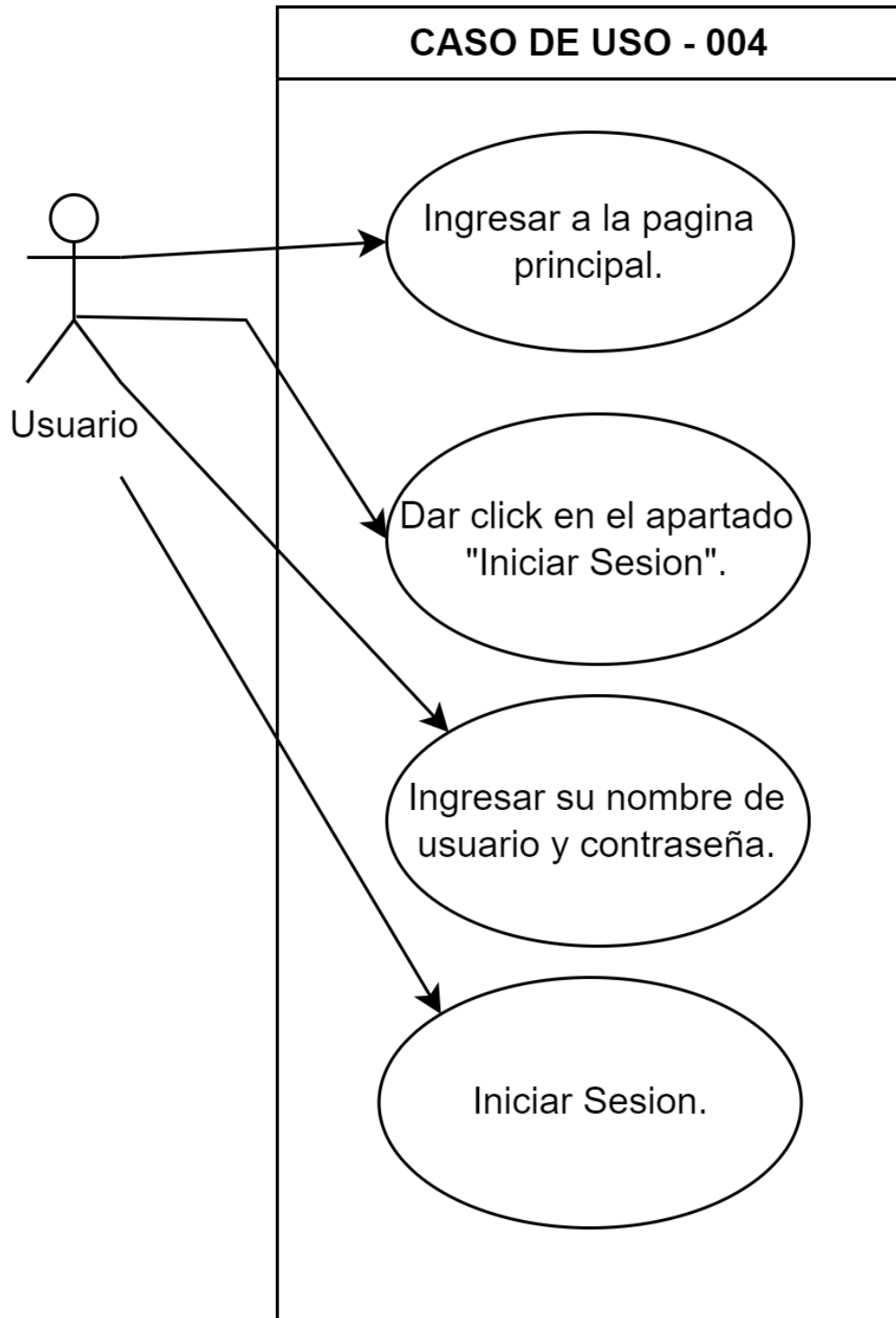


CASOS DE USO









CASOS DE USO EXTENDIDOS

Caso 0001. Describe el proceso para crear un Post.

- Actor Principal.
 - Usuario.
- Actores Involucrados e intereses.
 - Usuario: creación de Post.
- Precondiciones.
 - El usuario debe estar loggeado dentro del sistema.
- Garantías de Éxitos.
 - Se podrá crear un post dentro de la página.
- Escenario principal de éxito.
 - 1) El usuario se encuentra loggeado.
 - 2) El usuario se dirige al apartado para la creación de un nuevo Post.
 - 3) El usuario ingresa el contenido correspondiente.
 - 4) El usuario crea el Post.
- Extensiones.
 - 3a) El usuario no puede ingresar el contenido correspondiente.
- Requisitos especiales.
 - Interfaz de usuario fácil e intuitiva.
- Lista de tecnología y variación de datos.
 - 1a) El servidor se encuentra caído y el usuario no puede iniciar sesión.
 - 4a) La base de datos se encuentra caída y no se pueden hacer inserciones en la misma.
- Frecuencia.
 - Continuo.
- Temas abiertos.
 - ¿Qué sucede si el sistema se encuentra caído?

Caso 0002. Describe el proceso para comentar un Post.

- Actor Principal.
 - Usuario.
- Actores Involucrados e intereses.
 - Usuario: creación de un comentario dentro de un Post.
- Precondiciones.
 - El usuario debe estar loggeado dentro del sistema.
 - Debe haber un Post creado para poder comentarlo.
- Garantías de Éxitos.
 - Se podrá crear un comentario dentro de un Post en el sistema.
- Escenario principal de éxito.
 - 1) El usuario se encuentra loggeado.
 - 2) El usuario se dirige a su timeline para abrir un post.
 - 3) El usuario abre el Post en cuestión.
 - 4) El usuario lee el post y busca la parte inferior de la página.
 - 5) El usuario ingresa el contenido de su comentario en el contenedor.
 - 6) El usuario publica su post.
- Extensiones.
 - 3a) El usuario desea comentar un post que ya no existe.
 - El sistema le mostrará una alerta informando que el post ya no existe.
- Requisitos especiales.
 - Interfaz de usuario fácil e intuitiva.
- Lista de tecnología y variación de datos.
 - 2a) El servidor se encuentra caído y el usuario no puede navegar dentro del sistema.
 - 6a) La base de datos se encuentra caída y no se pueden hacer inserciones en la misma.
- Frecuencia.
 - Continuo.
- Temas abiertos.
 - ¿Qué sucede si el sistema se encuentra caído?
 - ¿Se puede reaccionar a los comentarios dentro un post?

Caso 0003. Describe el proceso para crear una cuenta dentro del sistema.

- Actor Principal.
 - Usuario.
- Actores Involucrados e intereses.
 - Usuario: creación de cuenta.
- Precondiciones.
 - El usuario debe tener acceso a internet.
- Garantías de Éxitos.
 - Se podrá crear una cuenta dentro del sistema.
- Escenario principal de éxito.
 - 1) El usuario cuenta con acceso a internet.
 - 2) El usuario se dirige al apartado para la creación de una nueva cuenta.
 - 3) El usuario ingresa los parámetros solicitados por el sistema.
 - 4) El usuario crea su cuenta.
- Extensiones.
 - 3a) El usuario no ingresa un dato importante, como su nombre, por ejemplo.
 - El sistema le mostrará una alerta donde se le informará que faltan datos obligatorios.
- Requisitos especiales.
 - Interfaz de usuario fácil e intuitiva.
- Lista de tecnología y variación de datos.
 - 1a) El servidor se encuentra caído y el usuario no puede acceder a la página.
 - 4a) La base de datos se encuentra caída y no se pueden hacer inserciones en la misma.
- Frecuencia.
 - Continuo.
- Temas abiertos.
 - ¿Qué sucede si el sistema se encuentra caído?
 - ¿Qué sucede si el usuario inserta datos falsos?

Caso 0004. Describe el proceso para poder loggarse dentro del sistema.

- Actor Principal.
 - Usuario.
- Actores Involucrados e intereses.
 - Usuario: login en el sistema.
- Precondiciones.
 - El usuario debe contar con acceso a internet.
- Garantías de Éxitos.
 - Se podrá ingresar al sistema.
- Escenario principal de éxito.
 - 1) El usuario se encuentra conectado a internet.
 - 2) El usuario se dirige al apartado de inicio de sesión.
 - 3) El usuario ingresa los parámetros solicitados.
 - 4) El usuario ingresa al sistema.
- Extensiones.
 - 3a) El usuario ingresa datos erróneos.
 - El sistema le solicitará que ingrese sus datos de nuevo, luego de 10 intentos fallidos, el sistema bloquea temporalmente su IP.
- Requisitos especiales.
 - Interfaz de usuario fácil e intuitiva.
- Lista de tecnología y variación de datos.
 - 1a) El servidor se encuentra caído y el usuario no puede iniciar sesión.
 - 4a) La base de datos se encuentra caída y no se pueden hacer consultas en la misma.
- Frecuencia.
 - Continuo.
- Temas abiertos.
 - ¿Qué sucede si el sistema se encuentra caído?
 - ¿Qué sucede si el sistema toma como erróneas sus credenciales, pero están bien?

DIAGRAMAS DE SECUENCIA

Diagrama de Secuencia Caso de Uso 001

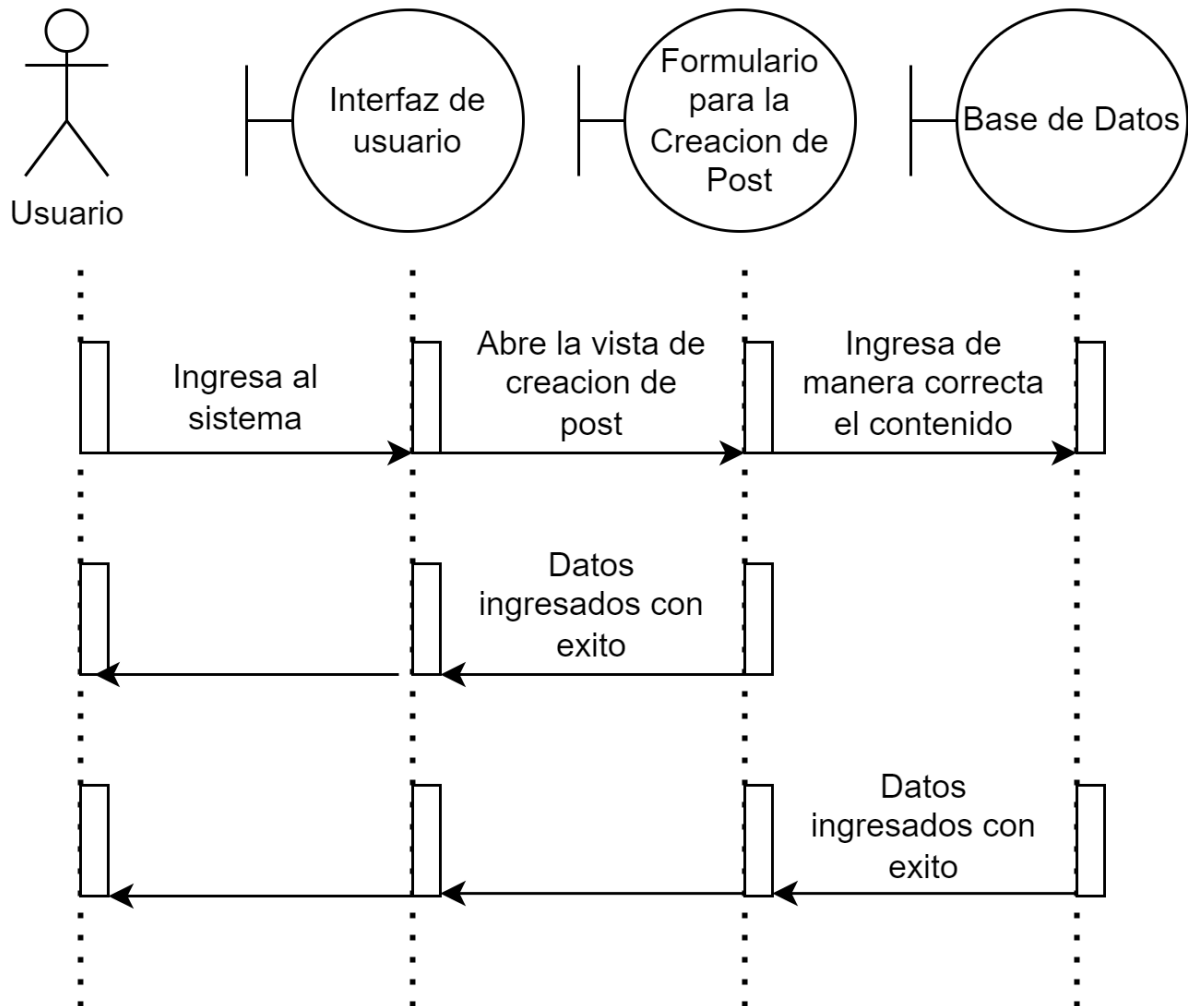


Diagrama de Secuencia Caso de Uso 002

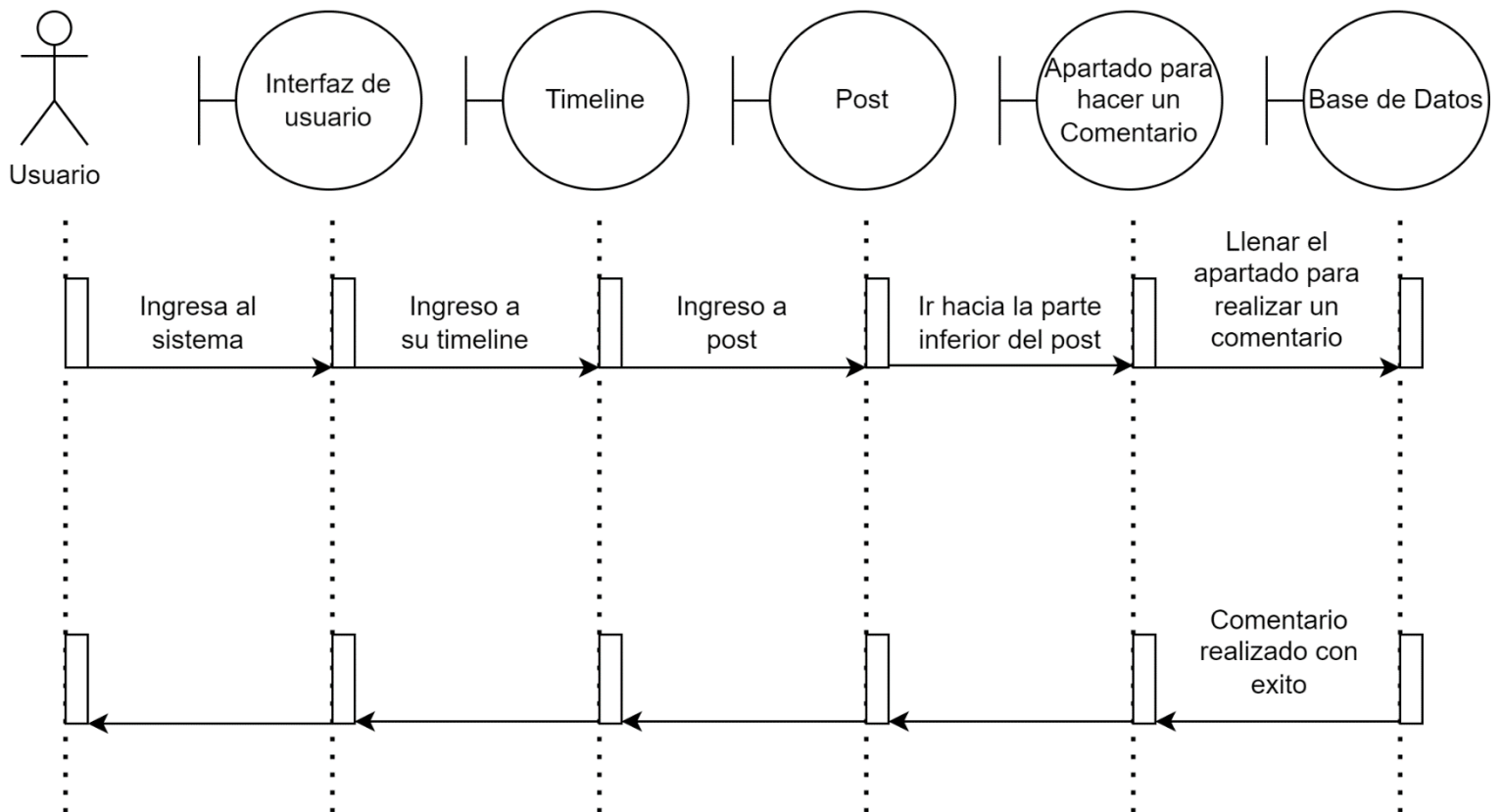


Diagrama de Secuencia Caso de Uso 003

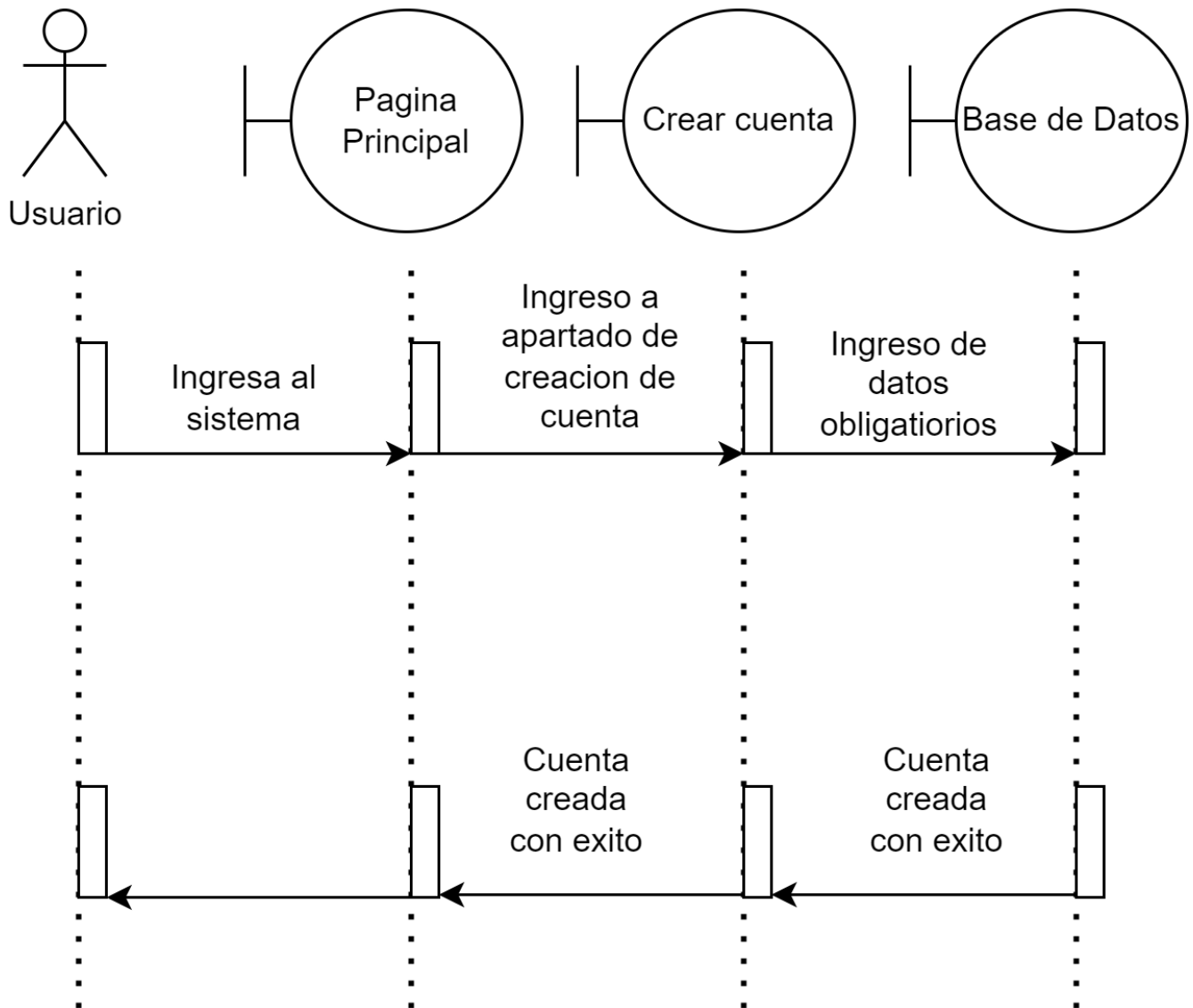
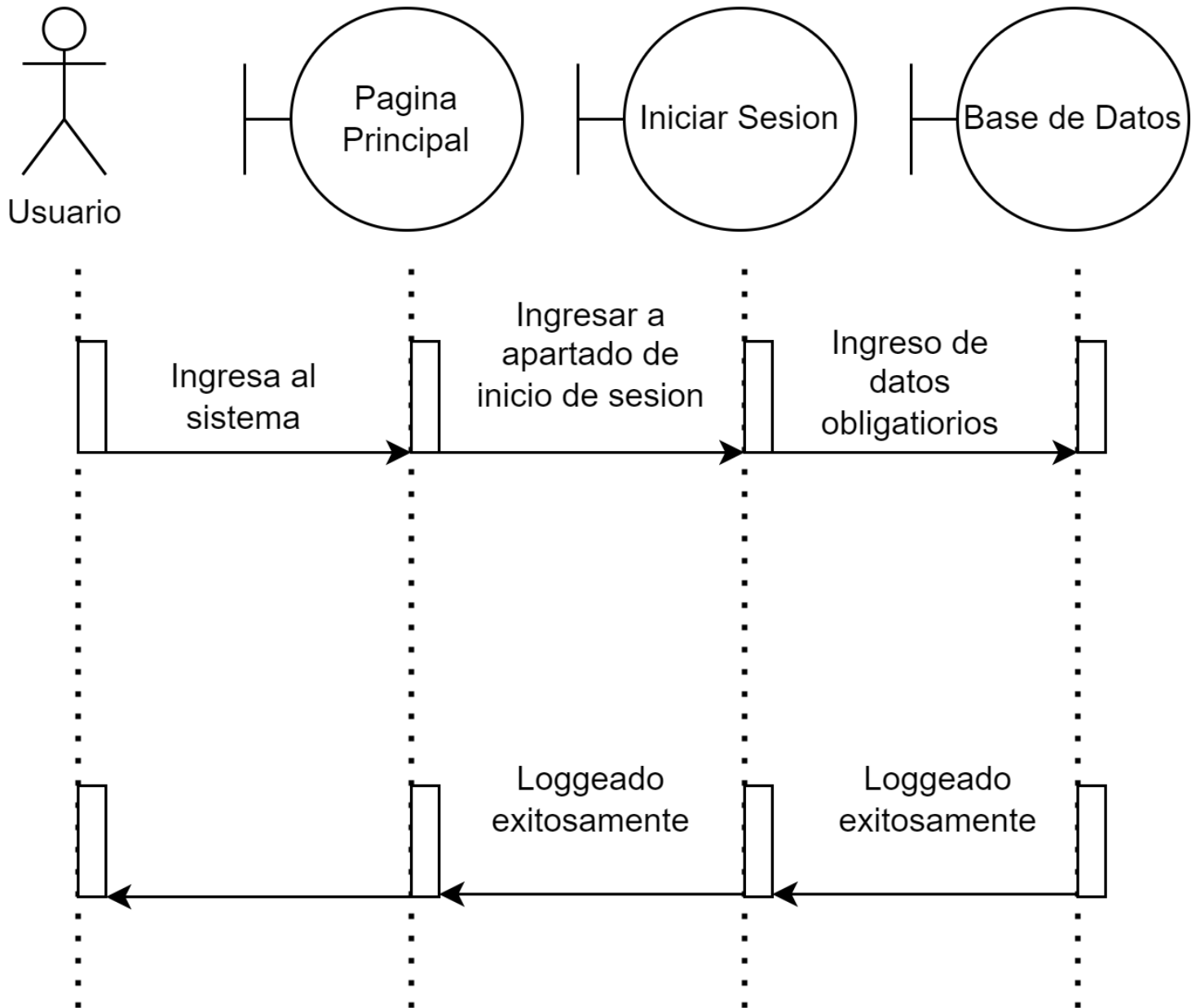


Diagrama de Secuencia Caso de Uso 004



PROCESOS DE INSTALACIÓN

Para poder testear el proyecto debemos tener instalados los siguientes programas en nuestro sistema. En este caso todo el entorno de trabajo esta configurado en Windows 10.

1. Xampp. (Contiene nuestro servidor Apache, MySQL, así como PHP).
2. VS Code.
3. Un navegador.

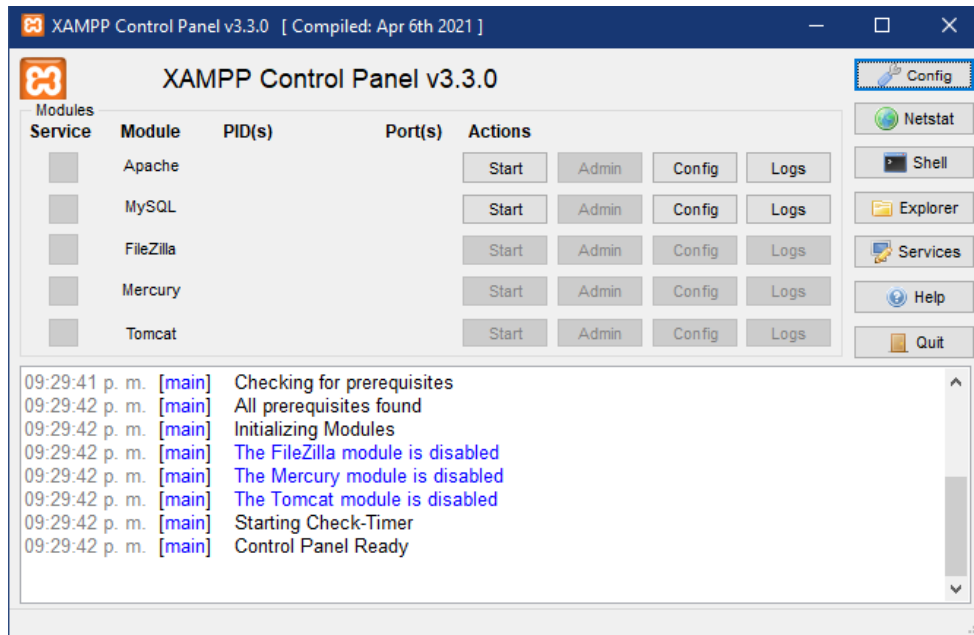
Puede referirse a las siguientes guías de instalación.

Xampp: <https://www.mclibre.org/consultar/php/otros/xampp-instalacion-windows.html>

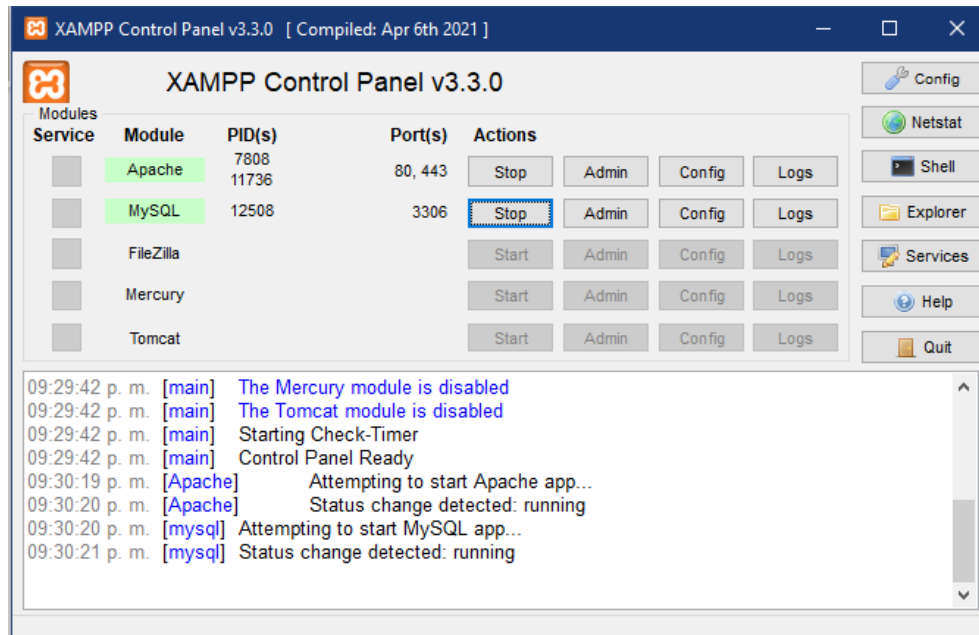
VS Code: <https://www.mclibre.org/consultar/informatica/lecciones/vsc-instalacion.html>

APLICACIÓN

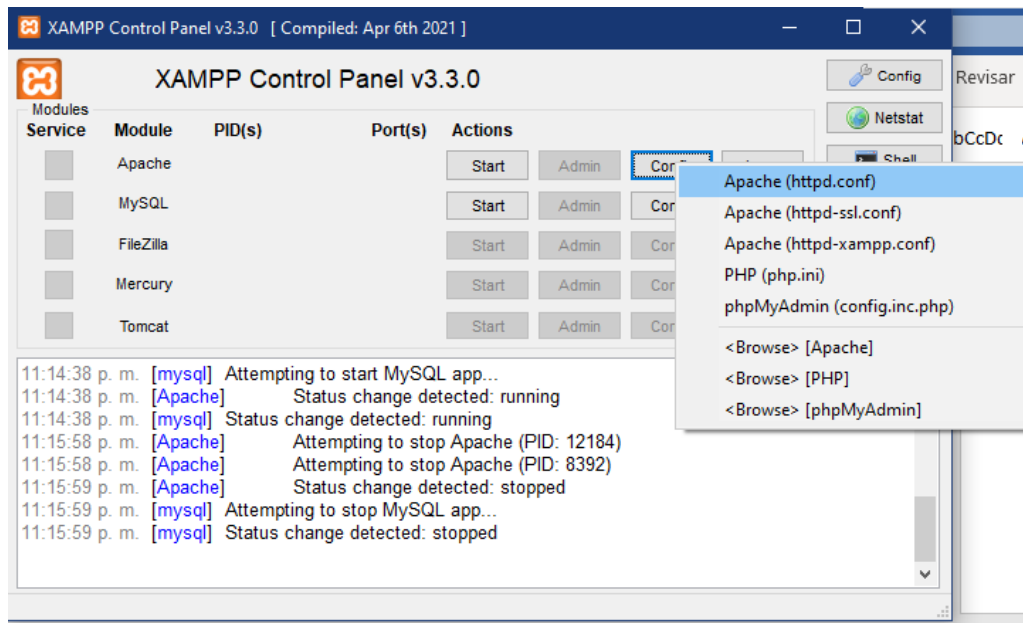
- Para iniciar la aplicación debe iniciar “Xampp Control Panel”, puede buscarlo en su buscador de Windows.



- Debe iniciar los servicios de Apache y MySQL.



- Un paso importante es cambiar nuestro archivo, httpd.conf



- Debemos buscar esta parte dentro del archivo y cambiarla por la ruta donde esta alojada nuestra aplicación. Esto lo hacemos para cambiar la ruta por defecto de apache.

```

httpd.conf: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:\Users\ACER\Desktop\Segundo Semestre 2023\Teoria de Sistemas I\Practica 1\BlogChilero"
<Directory "C:\Users\ACER\Desktop\Segundo Semestre 2023\Teoria de Sistemas I\Practica 1\BlogChilero">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #

```

- Debe ingresar la siguiente url: `http://localhost/login.html`

