**Explicación de Diagramas**

Los siguientes diagramas se realizaron con la herramienta para desarrollo Mermaid.

Página web: https://www.mermaidchart.com/app/projects

***1. Diagrama de componentes***

**Código:**

---

config:

  layout: dagre

---

**flowchart** TB

**subgraph** Cliente["Cliente (Frontend)"]

        A1["HTML<br><small>Estructura del contenido</small>"]

        A2["CSS / SASS<br><small>Estilos y diseño visual</small>"]

        A3["JavaScript<br><small>Lógica del lado cliente</small>"]

        A4["AJAX<br><small>Comunicación asíncrona con el servidor</small>"]

**end**

**subgraph** Servidor["Servidor (Backend)"]

        B1["PHP Puro<br><small>Lógica del sistema y controladores</small>"]

        B2["Patrón Active Record<br><small>Mapeo entre clases y tablas</small>"]

        B3["MySQL<br><small>Gestión y almacenamiento de datos</small>"]

        B4["Apache XAMPP<br><small>Servidor web local</small>"]

        B5["Gulp.js<br><small>Automatización de tareas</small>"]

        B6["Node.js + NPM<br><small>Entorno y gestión de dependencias</small>"]

**end**

    A1 **-->** A2

    A2 **-->** A3

    A3 **-->** A4

    A4 **-->** B1

    B1 **-->** B2 **&** B4

    B2 **-->** B3

    B5 **-->** B6

    Cliente **--** HTTP AJAX **-->** Servidor

    Servidor **--** Respuestas dinámicas **-->** Cliente

    title("Diagrama de Componentes - Arquitectura Cliente-Servidor (MVC)")

**Diagrama:**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Descripción del diagrama:**

* El **Cliente** contiene los componentes relacionados con la interfaz y la interacción del usuario.
* El **Servidor** incluye los elementos responsables de la lógica, la persistencia y el procesamiento de datos.
* Las flechas muestran la **comunicación bidireccional** entre el cliente y el servidor mediante peticiones HTTP o AJAX.
* También se refleja la **dependencia entre herramientas** (por ejemplo, Gulp requiere Node.js y NPM).

***2. Diagrama de clases***

**Código:**

**classDiagram**

direction TB

**class** Usuario **{**

**+**int id\_usuario

**+**string nombre

**+**string email

**+**AbrirPagina**()**

**+**CerrarPaginaweb**()**

**+**VerPripiedad**()**

**}**

**class** Propiedad **{**

**+**int id\_propiedad

**+**string titulo

**+**string descripcion

**+**string direccion

**+**float precio

**+**string tipo

**+**string estado

**+**string imagen

**+**float Habitaciones

**+**float baños

**+**float cochera

**+**publicar**()**

**+**editar**()**

**+**eliminar**()**

**}**

**class** Consulta **{**

**+**int id\_consulta

**+**string nombre

**+**string email

**+**float telefono

**+**string mensaje

**+**enviar**()**

**}**

**class** Administrador **{**

**+**int id\_admin

**+**string usuario

**+**string email

**+**gestionarPropiedades**()**

**+**responderConsultas**()**

**}**

**class** BaseDatos **{**

**+**conectar**()**

**+**ejecutarConsulta**()**

**+**cerrarConexion**()**

**}**

    Usuario **<|--** Administrador **:** hereda

    Propiedad "1" **-->** "0..\*" Consulta **:** recibe

    BaseDatos **-->** Usuario **:** CRUD

    BaseDatos **-->** Propiedad **:** CRUD

    BaseDatos **-->** Consulta **:** CRUD

**Diagrama:**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Descripción del diagrama:**

* **Usuario:** clase general que representa a cualquier persona que utiliza el sistema (cliente o administrador).
* **Administrador:** hereda de *Usuario* y tiene permisos adicionales para gestionar propiedades y usuarios.
* **Propiedad:** representa la información de los inmuebles (nombre, descripción, tipo, precio, etc.).
* **Consulta:** almacena los mensajes o formularios de contacto enviados por los usuarios interesados.
* **BaseDatos:** clase encargada de las operaciones de conexión y gestión de datos.

**¿Qué significa CRUD?**

El término **CRUD** es un acrónimo que describe las **cuatro operaciones básicas** que se pueden realizar sobre los datos en una aplicación o base de datos.

**Significado de CRUD**

| **Letra** | **Significado** | **Acción en el sistema** | **Acción en la base de datos** |
| --- | --- | --- | --- |
| **C** | **Create** | Crear o registrar un nuevo elemento (por ejemplo, una propiedad o usuario) | INSERT |
| **R** | **Read** | Leer o consultar información existente | SELECT |
| **U** | **Update** | Actualizar o modificar datos existentes | UPDATE |
| **D** | **Delete** | Eliminar un registro del sistema o base de datos | DELETE |

***3. Diagramas de secuencia***

**3.1 Secuencia: Enviar consulta desde pagina web**

**Código:**

**sequenceDiagram**

    title Diagrama de Secuencia - Envío de Consulta desde el Sitio Web

    participant Usuario as Usuario (Navegador)

    participant Vista as Vista (Formulario de contacto)

    participant Controlador as ControladorConsulta (PHP)

    participant Modelo as ModeloConsulta

    participant BD as Base de Datos (MySQL)

    %% Flujo principal

    Usuario**->>**Vista**:** Ingresa datos y presiona "Enviar"

    Vista**->>**Controlador**:** Envía datos del formulario (POST /consulta)

    Controlador**->>**Modelo**:** Llama a guardarConsulta(datos)

    Modelo**->>**BD**:** Ejecuta INSERT INTO consultas (...)

    BD**-->>**Modelo**:** Confirma inserción exitosa

    Modelo**-->>**Controlador**:** Retorna resultado (éxito)

    Controlador**-->>**Vista**:** Envía mensaje de confirmación

    Vista**-->>**Usuario**:** Muestra "Consulta enviada correctamente"

    %% Flujo alternativo (error)

    note over Modelo,BD**:** Si ocurre un error de conexión o validación

    Modelo**-->>**Controlador**:** Retorna error

    Controlador**-->>**Vista**:** Envía mensaje de error

    Vista**-->>**Usuario**:** Muestra "No se pudo enviar la consulta"

**Diagrama:**

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Descripción del diagrama:**

* **Usuario:** inicia el proceso desde el navegador.
* **Vista:** representa la interfaz del formulario en el sitio web.
* **Controlador:** recibe los datos, valida y gestiona la comunicación con el modelo.
* **Modelo:** contiene la lógica para guardar la información en la base de datos.
* **Base de Datos (MySQL):** almacena la información de la consulta.
* El diagrama muestra tanto el **flujo normal (éxito)** como el **flujo alternativo (error)**.

**3.2 Secuencia: Publicar una Propiedad**

**Código:**

sequenceDiagram

    title Diagrama de Secuencia - Publicación de una Propiedad

    participant Propietario as Propietario (Usuario)

    participant Vista as Vista (Formulario de publicación)

    participant Controlador as ControladorPropiedad (PHP)

    participant Modelo as ModeloPropiedad

    participant BD as Base de Datos (MySQL)

    %% Flujo principal

    Propietario->>Vista: Accede al formulario de nueva propiedad

    Vista-->>Propietario: Muestra campos para completar (título, descripción, precio, etc.)

    Propietario->>Vista: Completa los datos y presiona "Publicar"

    Vista->>Controlador: Envía los datos mediante POST /publicarPropiedad

    Controlador->>Controlador: Valida los datos ingresados

    Controlador->>Modelo: Llama a crearPropiedad(datos)

    Modelo->>BD: Ejecuta INSERT INTO propiedades (...)

    BD-->>Modelo: Devuelve confirmación de registro

    Modelo-->>Controlador: Retorna resultado exitoso

    Controlador-->>Vista: Envía mensaje de confirmación

    Vista-->>Propietario: Muestra "Propiedad publicada correctamente"

    %% Flujo alternativo (error de validación o base de datos)

    note over Controlador,BD: Si hay error en validación o conexión

    Controlador-->>Vista: Envía mensaje de error

    Vista-->>Propietario: Muestra "Error al publicar la propiedad"

**Diagrama:**

**Diagrama

El contenido generado por IA puede ser incorrecto.**

**Descripción del proceso:**

1. El **propietario** accede al formulario de publicación.
2. La **vista** presenta los campos requeridos (título, descripción, tipo, precio, imágenes, etc.).
3. Al enviar el formulario, los datos se envían al **controlador**, que realiza las validaciones necesarias.
4. Si los datos son correctos, el **controlador** invoca al **modelo**, que ejecuta la operación de inserción en la **base de datos MySQL**.
5. Una vez confirmada la inserción, el sistema devuelve una **respuesta exitosa** a la vista.
6. Si ocurre un error (por ejemplo, falta de datos o fallo en la base de datos), se muestra un **mensaje de error** al usuario.

***4. Diagramas de Actividad:***

**4.1 Actividad: Enviar Consulta**

**Código:**

**flowchart** TD

    A(["Inicio"]) **-->** B["Usuario ingresa al sitio web"]

    B **-->** C{"¿Consulta sobre una propiedad?"}

    C **--** Sí **-->** D["Usuario selecciona una propiedad"]

    D **-->** E["Sistema muestra detalles de la propiedad"]

    E **-->** F@{ label: "Usuario hace clic en 'Enviar consulta'" }

    C **--** No **-->** F

    F **-->** G["Sistema muestra formulario de contacto"]

    G **-->** H["Usuario completa los campos requeridos"]

    H **-->** I@{ label: "Usuario hace clic en 'Enviar'" }

    I **-->** J{"¿Datos válidos?"}

    J **--** No **-->** K["Mostrar mensaje de error"]

    K **-->** G

    J **--** Sí **-->** L{"¿Consulta vinculada a propiedad?"}

    L **--** Sí **-->** M["Guardar o enviar consulta asociada a la propiedad seleccionada"]

    L **--** No **-->** N["Guardar o enviar consulta general"]

    M **-->** O@{ label: "Mostrar mensaje de confirmación 'Consulta enviada con éxito'" }

    N **-->** O

    O **-->** P(["Fin"])

    F@{ shape: rect}

    I@{ shape: rect}

    O@{ shape: rect}

**Diagrama:**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Explicación del flujo:**

* El usuario puede iniciar una consulta general o sobre una propiedad.
* Si selecciona una propiedad, el sistema muestra sus detalles antes del formulario.
* Si no lo hace, puede acceder directamente al formulario.
* Luego el sistema valida los datos, guarda o envía la información según corresponda, y confirma el envío.

**4.2 Actividad: Publicar una Propiedad**

**Código:**

**flowchart** TD

    title("Diagrama de Actividades - Publicación de una Propiedad")

    %% Estados inicial y final

    A([Inicio]) **-->** B[Propietario accede al formulario de publicación]

    B **-->** C[Completa los datos de la propiedad]

    C **-->** D{Validar datos ingresados}

    %% Validación

    D **-->**|Datos válidos| E[Enviar datos al servidor]

    D **-->**|Datos inválidos| F[Mostrar mensaje de error al usuario]

    F **-->** C

    %% Procesamiento en el servidor

    E **-->** G[Controlador recibe y procesa los datos]

    G **-->** H[Llamar al modelo para crearPropiedad]

    H **-->** I{Inserción exitosa en la base de datos?}

    %% Resultado

    I **-->**|Sí| J[Mostrar mensaje 'Propiedad publicada correctamente']

    I **-->**|No| K[Mostrar mensaje 'Error al publicar la propiedad']

    %% Fin del flujo

    J **-->** L([Fin])

    K **-->** L

**Diagrama:**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Descripción del flujo:**

1. El proceso comienza cuando el **propietario accede al formulario de publicación**.
2. Completa los datos requeridos (título, descripción, precio, tipo, imágenes, etc.).
3. El sistema **valida la información ingresada**.
   * Si los datos son **inválidos**, se muestra un mensaje de error y el usuario puede corregirlos.
   * Si son **válidos**, los datos se envían al servidor.
4. El **controlador** procesa la solicitud y llama al **modelo**, que intenta insertar la propiedad en la **base de datos**.
5. Si la inserción es exitosa, se muestra un mensaje confirmando la publicación.
6. Si ocurre un error (por ejemplo, de conexión o duplicado), se notifica al usuario.
7. Finalmente, el proceso concluye.