**UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA**

**UNAN – LEÓN**

****

**AÑO LECTIVO: 2025**

**SEMESTRE: II**

**Componente: Análisis y diseño de sistemas de información.**

**Tema:**

**Grupo:** 1

**Nombre del docente: William**

**Proyecto de encuestas.**

**Integrantes:**

*Ismael Alberto Martínez Lacayo 22-06305-0*

*Kevin Alejandro Sánchez Machado 22-01062-0*

*Samuel Antonio González Flores 22-05462-0*

León, Nicaragua, 2025.

**Visión General del Proyecto**

Este proyecto es un sistema de encuestas web, desarrollado con una arquitectura de frontend y backend para gestionar respuestas, catálogos y permisos de usuarios. La interfaz de usuario (index.html y styles.css) permite a los usuarios interactuar con la aplicación, mientras que el backend (db.php) maneja las operaciones de la base de datos (CRUD). La base de datos es MySQL y su estructura está definida en el archivo encuesta\_xvii.sql.

El proyecto está diseñado para ser funcional y fácil de usar, con un enfoque en la interacción directa con una base de datos a través de una API PHP simple, aunque se advierte que no es seguro para un entorno de producción.

**Estructura del Archivo**

El proyecto está organizado en los siguientes archivos:

* **index.html**: El archivo principal que contiene la **estructura de la interfaz de usuario (UI)** y la lógica del frontend. Incluye elementos como el formulario de registro, la tabla de respuestas, las vistas para catálogos y la matriz de permisos.
* **styles.css**: Contiene todo el **estilo y la presentación visual** de la aplicación. Define los colores, la tipografía, el diseño (usando Grid y Flexbox) y los efectos visuales para una experiencia de usuario atractiva.
* **script.js**: Este archivo, aunque no lo proporcionaste, es el código JavaScript que normalmente manejaría la **lógica del frontend**, como la validación de formularios, las peticiones HTTP al backend, la manipulación del DOM para mostrar datos y la gestión de eventos de la UI. Sin este archivo, la interfaz no tendría interactividad.
* **db.php**: El **backend de la aplicación**. Es una API simple escrita en PHP que actúa como intermediario entre el frontend y la base de datos MySQL. Procesa las solicitudes del frontend (como guardar una encuesta o listar respuestas) y realiza las operaciones correspondientes en la base de datos.
* **encuesta\_xvii.sql**: El script SQL que define la **estructura de la base de datos** y las tablas necesarias para el proyecto. Este archivo contiene las sentencias CREATE TABLE para las encuestas, catálogos, usuarios y roles, así como datos de ejemplo para iniciar el proyecto.

**Descripción de la Funcionalidad**

La aplicación cuenta con varias secciones, accesibles a través de un sistema de pestañas:

1. **Encuestas**: Esta es la vista principal. Permite a los usuarios **registrar nuevas encuestas** y **editar o eliminar** las existentes. Las respuestas se muestran en una tabla con opciones de búsqueda, y se pueden exportar a formatos CSV y PDF.
2. **Catálogos**: Muestra el contenido de todas las tablas de catálogo que se usan en los menús desplegables del formulario de la encuesta (por ejemplo, sexo, departamento, facultad, etc.).
3. **Usuarios y Roles**: Permite visualizar los usuarios y sus roles. La característica más importante de esta sección es la **matriz de permisos**, que muestra qué permisos tiene asignado cada rol. Esto es fundamental para la seguridad y la funcionalidad de la aplicación.
4. **Conexión**: Ofrece una herramienta de prueba para verificar la conexión con la base de datos a través de db.php y permite ver los datos en su formato JSON sin procesar.

**Base de Datos y Lógica del Backend (db.php & encuesta\_xvii.sql).**

El corazón del proyecto reside en la interacción con la base de datos MySQL.

* **encuesta\_xvii.sql**: Define las siguientes tablas:
  + **encuesta**: Almacena los datos principales de las encuestas, con campos para nombre, apellido, identificación, etc.
  + **Tablas de catálogo (e.g., xvii, xvi)**: Contienen las opciones para los menús desplegables del formulario. La lógica está diseñada para que el frontend pueda cargar dinámicamente estas opciones.
  + **rol, permiso, rol\_permiso**: Estas tablas establecen un sistema de roles y permisos. Un rol puede tener múltiples permisos y un permiso puede estar asignado a múltiples roles.
  + **usuario**: Almacena la información de los usuarios (login, clave, rol).
* **db.php**: Este script PHP gestiona las solicitudes de la siguiente manera:
  + Establece una conexión a la base de datos MySQL usando la extensión mysqli.
  + Utiliza un switch para manejar las diferentes acciones (action) solicitadas por el frontend, como:
    - list\_respuestas: Devuelve todas las encuestas en formato JSON.
    - create\_respuesta: Inserta una nueva encuesta en la tabla encuesta.
    - update\_respuesta: Actualiza una encuesta existente.
    - delete\_respuesta: Elimina una encuesta.
    - list\_catalogo: Devuelve los datos de una tabla de catálogo específica.
  + La comunicación se realiza a través de **peticiones GET y POST**, y las respuestas se envían en formato JSON.

**Análisis de cada archivo.**

**📄 Análisis del archivo db.php**

Este archivo PHP actúa como el **backend de la aplicación**, una API que conecta el frontend con la base de datos MySQL. Su función principal es procesar las solicitudes de datos (lectura, creación, actualización y eliminación) desde el navegador.

* **Conexión a la Base de Datos:** Se conecta directamente a la base de datos MySQL usando la extensión mysqli. Se definen las credenciales (host, user, pass, db) al inicio del script. El código incluye un manejo básico de errores de conexión y establece el charset a utf8mb4.
* **Manejo de Solicitudes:** Utiliza una estructura switch para gestionar diferentes "acciones" solicitadas a través de la URL (parámetro action). Por ejemplo, db.php?action=list\_respuestas activa la función para listar encuestas, y db.php?action=create\_respuesta para crear una nueva.
* **Operaciones CRUD:**
  + list\_respuestas: Consulta todas las encuestas de la tabla respuestas y las devuelve como un arreglo JSON.
  + create\_respuesta: Recibe los datos de una nueva encuesta en formato JSON, utiliza una **sentencia preparada** (prepare()) para insertar los datos en la tabla respuestas, y devuelve el nuevo registro.
  + update\_respuesta: Recibe un número de encuesta y los datos a modificar, y actualiza el registro correspondiente en la base de datos.
  + delete\_respuesta: Elimina una encuesta basándose en su número.
* **Catálogos y Permisos:** También maneja solicitudes para listar los datos de las tablas de catálogo (como iii\_sexo o v\_departamento) y las tablas de usuarios, roles y permisos.

A screenshot of a computer

AI-generated content may be incorrect.

**Análisis del Diagrama de Entidad-Relación**

El diagrama representa la estructura de la base de datos del proyecto de encuestas, mostrando las tablas, sus columnas y las relaciones entre ellas. La base de datos está bien normalizada, lo que significa que los datos se almacenan de manera organizada para evitar redundancia y mejorar la integridad.

Tablas Principales

1. respuestas: Esta es la tabla central. Almacena las respuestas de cada encuesta. Tiene una clave primaria (PRIMARY KEY), numero, que identifica de manera única cada respuesta. Sus campos de I a XVIII corresponden a las preguntas del formulario.
   * Normalización: En lugar de guardar los valores de texto directamente para preguntas con opciones predefinidas, esta tabla utiliza claves foráneas (FOREIGN KEY) que se vinculan a las tablas de catálogo. Por ejemplo, el campo III\_sexo (id\_sexo) no almacena "Masculino" o "Femenino", sino un número que se refiere a la clave en la tabla iii\_sexo. Esta práctica es crucial para mantener la consistencia de los datos.
2. Tablas de Catálogo: Un grupo de tablas diseñadas para almacenar las opciones de los menús desplegables del formulario. Cada una tiene dos campos:
   * clave (id): Un identificador numérico único.
   * valor: La descripción textual de la opción.

Ejemplos de estas tablas son v\_departamento, vii\_facultad, x\_matricula, y todas las que terminan en \_catalogo. Esto asegura que los datos sean coherentes y facilita su mantenimiento.

Tablas de Gestión de Usuarios y Permisos

1. usuario: Almacena la información de los usuarios que pueden acceder al sistema. Su clave primaria es idusuario. El campo idrol es una clave foránea que la conecta con la tabla rol, asignando un rol específico a cada usuario.
2. rol: Contiene los roles disponibles en el sistema (idrol, nombre\_rol). Los roles definen el nivel de acceso y las funcionalidades que un usuario puede tener, como administrador o estudiante.
3. permiso: Enumera las acciones específicas que se pueden realizar en la aplicación (idpermiso, accion). Por ejemplo, registrar\_encuesta o borrar\_encuesta.
4. rol\_permiso: Esta es una tabla de unión que conecta los roles con los permisos. Es necesaria porque la relación entre roles y permisos es de muchos a muchos (un rol puede tener varios permisos y un permiso puede estar asociado a varios roles). Su clave primaria compuesta está formada por idrol e idpermiso.

Relaciones Clave

El diagrama muestra claramente cómo las tablas se relacionan entre sí:

* Uno a Muchos (1:N):
  + usuario y rol: Un rol puede ser asignado a muchos usuarios, pero cada usuario tiene un solo rol.
  + respuestas y las tablas de catálogo: Una opción de un catálogo (por ejemplo, "Ingeniería en Sistemas") puede estar en muchas respuestas, pero cada campo en una respuesta solo puede vincularse a una única opción.
* Muchos a Muchos (N:M):
  + rol y permiso: Un rol tiene muchos permisos, y un permiso puede pertenecer a muchos roles. La tabla rol\_permiso se encarga de gestionar esta relación, evitando la duplicación de datos.

**💾 Análisis del archivo encuesta\_xvii.sql**

Este archivo es el **esquema completo de la base de datos MySQL** y contiene todas las sentencias necesarias para crear las tablas y poblar algunas con datos iniciales (dumping data).

* **Tabla respuestas:** Es la tabla principal que almacena los datos de la encuesta. Tiene campos como numero (clave primaria y autoincremental), I (nombre), II (apellido), IV (identificación) y una serie de campos numéricos (III al XVII) que se relacionan con las tablas de catálogo.
* **Tablas de Catálogo:** Se usan para almacenar los valores de los menús desplegables del formulario. Cada tabla de catálogo tiene los campos clave (un número) y valor (la descripción). Ejemplos de estas tablas son iii\_sexo, v\_departamento, vi\_ciudad, vii\_facultad, etc.. La tabla respuestas utiliza **claves foráneas** (FOREIGN KEY) para enlazar a estas tablas de catálogo.
* **Sistema de Roles y Permisos:**
  + rol: Define los roles de usuario (ej. admin, estudiante, profesor).
  + permiso: Enumera las acciones que se pueden realizar en la aplicación (ej. registrar\_encuesta, actualizar\_encuesta, exportar\_excel).
  + rol\_permiso: Es una tabla de unión que asigna permisos a roles específicos.
* **Tabla usuario:** Almacena los usuarios, su clave, nombre y el rol al que pertenecen.

**💻 Análisis del archivo index.html**

Este es el **esqueleto de la interfaz de usuario**. Define la estructura y el contenido de la página web que los usuarios ven e interactúan.

* **Estructura Base:** Incluye la declaración <!doctype html>, las etiquetas head (para metadatos, título, CSS y JavaScript) y body.
* **Encabezado (header):** Contiene la marca del proyecto (brand), un logo SVG y la barra de navegación principal con pestañas (puedes ver que las pestañas se controlan por el atributo data-view). También muestra información sobre el estado de la conexión a la base de datos, el rol del usuario y los permisos asignados en unos "chips".
* **Secciones de Contenido (main):** El contenido principal se divide en secciones (<section>) con IDs como view-encuestas, view-catalogos, etc., lo que permite que el JavaScript muestre u oculte el contenido según la pestaña seleccionada.
* **Formulario de Encuesta:** La sección de encuestas tiene un formulario (<form id="formEncuesta") que incluye campos de texto (<input>) y menús desplegables (<select>) para cada pregunta de la encuesta, con nombres de campo que coinciden con las columnas de la tabla respuestas.
* **Tablas de Datos:** Contiene una tabla para mostrar las respuestas de las encuestas (<table id="tablaRespuestas">), una para los usuarios (<table id="tablaUsuarios">) y la matriz de permisos.
* **Ventana de Login (dialog):** Hay un cuadro de diálogo para iniciar sesión (<dialog id="dlgLogin">) que utiliza un formulario para autenticar a los usuarios.

**📜 Análisis del archivo script.js**

Este es el **cerebro del frontend**, el código JavaScript que maneja toda la lógica del lado del cliente y la interacción con el usuario.

* **Configuración y Variables:** Define si se usa o no el backend de PHP (USE\_BACKEND) y un caché para los datos de catálogos (cacheCats). También inicializa variables para almacenar los datos de usuarios, roles y permisos.
* **Peticiones a la API:** La constante API contiene una serie de funciones async que realizan peticiones fetch a db.php para todas las operaciones (listar, crear, actualizar, borrar).
* **Lógica de Permisos:** La función hasPerm(permName) es crucial. Consulta las variables de roles y permisos cargadas del backend para determinar si el usuario actual tiene el permiso necesario para realizar una acción (ej. actualizar\_encuesta o borrar\_encuesta). Esto se utiliza para deshabilitar botones en la interfaz (guardiaUI()).
* **Manipulación del DOM:** Las funciones renderTabla, fillSelect y fillCatalogLists se encargan de cargar y mostrar los datos de la base de datos en las tablas y menús desplegables del index.html.
* **Manejo de Eventos:** El script escucha eventos como el clic en los botones de "editar" o "borrar", la presentación del formulario, y la interacción entre las pestañas.
* **Manejo de Datos Localmente:** El código está diseñado para funcionar también sin el backend (si USE\_BACKEND es false), almacenando los datos en el localStorage del navegador.

**🎨 Análisis del archivo styles.css**

Este archivo es responsable de toda la **apariencia visual y el diseño** de la aplicación.

* **Variables CSS (:root):** Define un conjunto de variables para los colores, lo que facilita la consistencia y la modificación de la paleta de colores de toda la aplicación. Hay variables para el fondo, el texto, colores de estado (danger, ok), bordes, etc..
* **Estilo General:** Establece estilos globales para el body (fuente, color de fondo degradado) y otros elementos como el box-sizing y la altura del cuerpo.
* **Diseño de Componentes:** Define el estilo para elementos específicos de la UI:
  + **Encabezado (header):** Tiene un fondo con efecto *blur* y se mantiene en la parte superior de la página (position: sticky).
  + **Navegación:** Los estilos de las pestañas (.tab) son definidos con transiciones para un efecto visual al pasar el cursor o al seleccionarlas.
  + **Botones (.btn):** Define el estilo para los botones con diferentes variantes (.ghost, .danger) y efectos de sombra y hover.
  + **Tarjetas (.card):** Define el estilo para los contenedores de contenido.
  + **Diseño con Grid:** El diseño principal utiliza display: grid con un sistema de columnas para organizar los elementos de la interfaz, como el formulario de la encuesta.
* **Media Queries:** Incluye una media query para adaptar el diseño a pantallas más pequeñas (@media (max-width:980px)), haciendo que el diseño de cuadrícula de 2 o 3 columnas se convierta en una sola columna.

**Análisis del Formulario de Encuesta**

El formulario de la encuesta está diseñado para recopilar información detallada sobre los encuestados, específicamente en un contexto académico o institucional. Los campos son claros y están organizados en dos columnas para una mejor visualización.

Los campos están categorizados de la siguiente manera:

* **Identificación Personal:** Los campos Nombre (I), Apellido (II) e Identificación (IV) son de texto libre. Esto permite la entrada de datos básicos del encuestado. El campo Sexo (III) es un menú desplegable, lo que asegura que la información se ingrese de manera estandarizada y se pueda usar para análisis de datos más limpios.
* **Información Académica y de Localización:** Los campos Departamento (V), Ciudad (VI), Facultad (VII) y Carrera (VIII) son menús desplegables. Esto indica que la aplicación depende de **tablas de catálogo** en la base de datos para cargar opciones predefinidas. Esta es una excelente práctica para mantener la consistencia de los datos y evitar errores de escritura.
* **Datos Adicionales y Cuantitativos:** Los campos del IX al XVII recopilan información más específica, como:
  + **Matrícula (X) y Becado (XI):** Menús desplegables que probablemente se vinculan a opciones como "Nuevo Ingreso", "Reingreso", "Externo", "Interno", etc.
  + **Campos de Valoración (XII, XIII, XIV):** Preguntas que evalúan aspectos como "Siempre", "Algunas Veces" o "Excelente", "Malo". Esto sugiere que se usan escalas de valoración para medir opiniones o hábitos.
  + **Motivo (XV), Gasto (XVI), Mejora esperada (XVII):** Estos campos capturan información sobre la motivación, los gastos relacionados con los estudios y las sugerencias de mejora.

El formulario cuenta con botones para **"Guardar"**, **"Cancelar"** y **"Cargar ejemplo"**. Los botones "Guardar" y "Cancelar" son estándar para la gestión de formularios. El botón "Cargar ejemplo" es una funcionalidad útil para fines de demostración y pruebas, ya que rellena el formulario automáticamente con datos de prueba, lo que acelera el desarrollo y la verificación.

El formulario está diseñado para ser intuitivo y eficiente, utilizando menús desplegables para guiar al usuario y garantizar la precisión de los datos. La estructura de dos columnas optimiza el uso del espacio en la pantalla y mejora la experiencia del usuario.

A screenshot of a computer

AI-generated content may be incorrect.