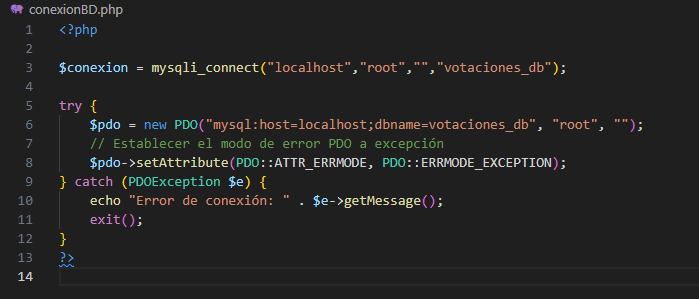
Documentación de la Conexión a la Base de Datos.

* Archivo: conexionBD.php

Este archivo es como el "cable" que conecta nuestra aplicación PHP con la base de datos de votaciones. Utilizamos dos métodos diferentes para hacer esta conexión, uno llamado MySQLi y otro PDO (que es un poco más moderno y elegante).



Conexión mediante MySQLi:

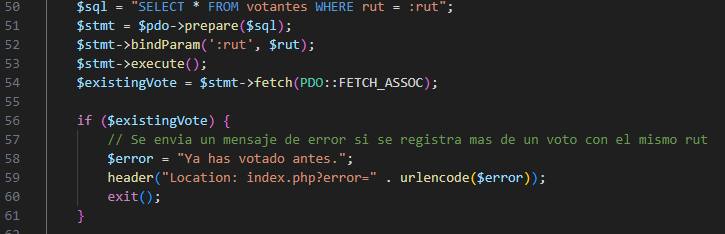
Usamos la función mysqli\_connect() para conectarnos a la base de datos, luego le pasamos los datos necesarios, como dónde está el servidor (localhost), el nombre de usuario (root), la contraseña (en este caso, no hay contraseña) y el nombre de la base de datos (votaciones\_db). Esta conexión permite hacer consultas a la base de datos utilizando MySQLi.

Conexión mediante PDO:

Aquí creamos lo que llamamos una "instancia" de la clase PDO para conectarnos a la base de datos, le decimos a PDO dónde está el servidor, el nombre de usuario, la contraseña y el nombre de la base de datos al igual que MySQLi, solo que esta vez utilizamos PDO para que antes de insertar un nuevo voto en la base de datos, usamos PDO para verificar si ya existe un voto registrado con el mismo RUT. Esto nos ayuda a prevenir la duplicación de votos que es una de las validaciones solicitadas.

La consulta SQL se prepara y ejecuta usando PDO para buscar cualquier registro existente con el RUT proporcionado.

En la siguiente imagen se puede visualizar parte del código del archivo procesar\_formulario.php, que muestra lo mencionado anteriormente.



Documentación de CSS.

* Archivo: estilos.css

He diseñado y aplicado estilos CSS para asegurar que el formulario sea fácil de usar y acorde a lo solicitado.

Pantalla de computadora con letras

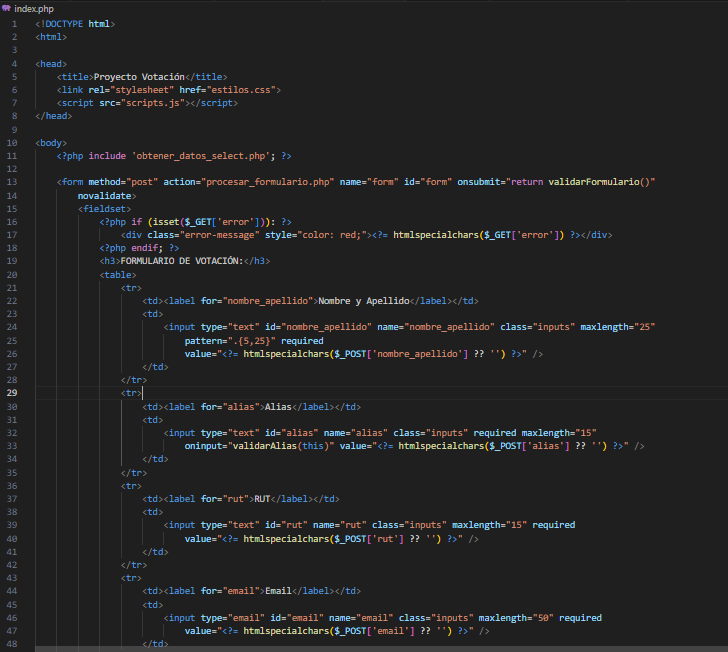
Descripción generada automáticamente con confianza media

* fieldset - Proporciona una estructura clara y centrada para el formulario.
* form - Centra el contenido del formulario y añade separación desde el borde superior.
* #checkbox-group – Separa las opciones del checkbox.
* .inputs – Ajusta el ancho de los inputs para adaptarse a diferentes tamaños de la pantalla.
* table – Proporciona suficiente espacio para los labels e inputs dentro del formulario.
* td – Añade un espacio alrededor del contenido en cada celda.
* #btn-votar – Separa el botón votar de otros elementos del formulario.
* .error-message – Descata los mensajes de error en color rojo.

Documentación de index.php.

* Archivo: index.php

Este archivo contiene el formulario de votación donde los usuarios pueden ingresar sus datos y emitir un voto.



Texto

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

El archivo index.php configura y presenta el formulario de votación para el proyecto, permitiendo a los usuarios ingresar sus datos personales y seleccionar su voto. Utiliza métodos POST para enviar los datos a procesar\_formulario.php y valida la información tanto en el cliente como en el servidor. Incluye manejadores de errores que muestran mensajes claros en caso de problemas y despliega listas dinámicas de regiones, comunas y candidatos obtenidos de la base de datos. La estructura está diseñada para ser intuitiva y garantizar que todos los campos necesarios se completen correctamente antes del envío, mejorando la experiencia del usuario y asegurando la integridad de los datos recolectados.

Documentación de obtener\_datos\_select.php.

* Archivo: obtener\_datos\_select.php

El archivo obtener\_datos\_select.php cumple la función de facilitar la obtención de datos de la base de datos para el formulario de votación. Incluye un enlace a conexionBD.php para establecer la conexión con la base de datos. Presenta una función llamada obtenerValores que se encarga de recuperar los registros de las tablas regiones, comunas y candidatos, almacenándolos en arreglos respectivos. Estos datos son esenciales para llenar los selectores del formulario dinámicamente. Al finalizar su ejecución, el archivo se encarga de cerrar la conexión a la base de datos para garantizar una gestión adecuada de los recursos.

Texto

Descripción generada automáticamente

Documentación de procesar\_formulario.php.

* Archivo: procesar\_formulario.php

El archivo procesar\_formulario.php se encarga de procesar los datos enviados desde el formulario de votación. Inicia una sesión para manejar mensajes de error y éxito. Contiene una función validarRUT para verificar la validez del RUT ingresado. Luego, comprueba si se ha enviado el formulario mediante el método POST y realiza la validación de los datos recibidos, como nombre, alias, RUT, correo electrónico, entre otros. Además, verifica que no exista un voto previo registrado con el mismo RUT en la base de datos. Posteriormente, consulta y asigna los nombres de región, comuna y candidato correspondientes a los IDs proporcionados. Si los datos son válidos y no hay duplicación de votos, realiza una inserción en la base de datos.

Adicionalmente a esto también se incorporó una función para que solo se pueda ingresar o seleccionar mejor dicho una comuna, cuando primero se haya seleccionado una región, así cada comuna aparecerá dependiendo la región que se seleccione.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Documentación de scripts

* Archivo: scripts.js

Este apartado detalla los scripts utilizados para las funciones de validación en su mayoría.

Se verifica si regionSelect tiene un valor diferente de null o undefined. Esto se hace para asegurarse de que el elemento fue encontrado en el DOM. Si existe, se añade un event listener al campo de selección de región que escucha el evento change, que se activa cuando el valor de la selección cambia.

Dentro de este event listener, se obtiene el valor seleccionado de la región (selectedRegion) y se seleccionan todas las opciones de comuna del campo de selección de comuna. Se recorren estas opciones y se verifica si el atributo data-region de cada opción coincide con la región seleccionada o si la opción es la opción por defecto (con valor ""). Si coincide, se muestra la opción; de lo contrario, se oculta.

Después de recorrer todas las opciones, se reinicia la selección de comuna (comunaSelect.value = "") y se deshabilita el campo de selección de comuna si no se ha seleccionado ninguna región (comunaSelect.disabled = selectedRegion === ""). Por último, si se seleccionó una región, se habilita el campo de selección de comuna (comunaSelect.removeAttribute("disabled")).

Texto

Descripción generada automáticamente

En las siguientes imágenes se detallan las diferentes validaciones para los campos de nombre\_apellido, rut, alias, región, comuna, candidato, email y los medios por los cuales se “entero de nosotros”, incluyendo variables para manejar los errores para estas.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Aquí está el código que valida el formato de RUT chileno.

Texto

Descripción generada automáticamente