****

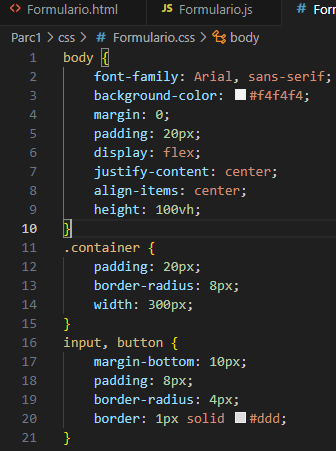
**Centro de Estudios Tecnológicos y Servicios Industriales Cetís No. 84**

**Gabriel Ignacio China Cortez**

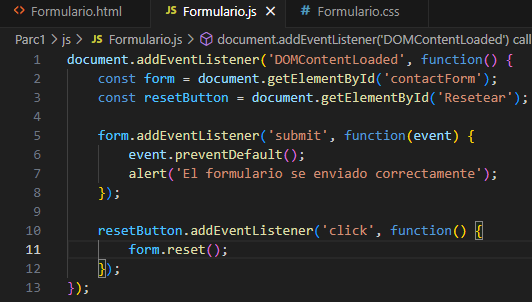
**Farías Cárdenas Javier Geovany**

Código

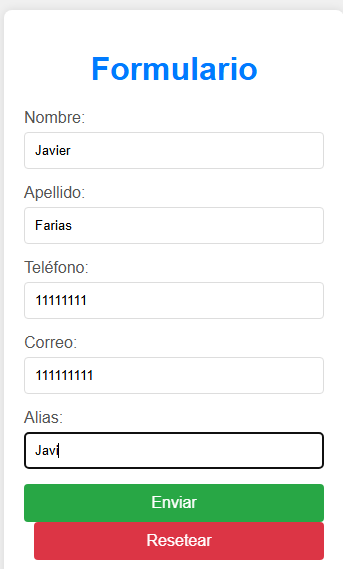
Css:

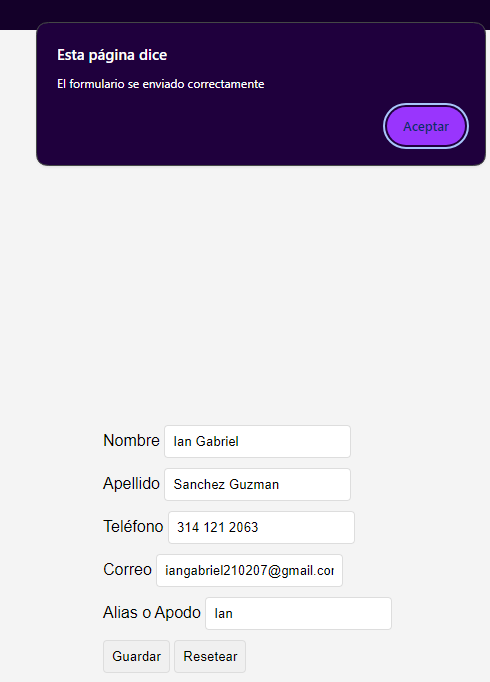


Js:

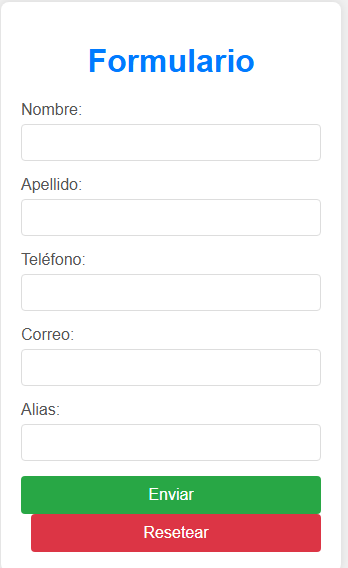


Funcionamiento:





Aquí es cuando le di al botón (Resetear)



5 buenas prácticas de HTML:

Organización del Código y Nombres Claros: Usa etiquetas como <header>, <nav>, <article>, y <footer> para hacer que tu código sea más claro y estructurado. Esto ayuda a que otros (y tú mismo en el futuro) entiendan mejor la organización de tu contenido.

Validación y Pruebas: Asegúrate de que tu HTML esté bien escrito y libre de errores usando herramientas como el W3C HTML Validator. Esto te ayudará a evitar problemas de compatibilidad entre diferentes navegadores.

Optimización del Rendimiento: Mantén tu HTML limpio, quitando cualquier código innecesario. Usa imágenes optimizadas y archivos multimedia de tamaño adecuado para que tu página cargue más rápido.

Mantenimiento y Reutilización de Código: Utiliza plantillas y componentes que puedas reutilizar para evitar escribir el mismo código varias veces. Esto hará que el mantenimiento sea mucho más sencillo.

Seguridad: Asegúrate de validar los datos ingresados por los usuarios tanto en el navegador como en el servidor, y usa HTTPS para mantener segura la transferencia de datos.

CSS:

Organización del Código y Nombres Claros: Usa nombres claros para tus clases e identificadores y organiza tu CSS con comentarios. Esto te ayudará a mantener el código ordenado y fácil de entender.

Validación y Pruebas: Revisa tu CSS con herramientas como CSS Validator para asegurarte de que no tenga errores que puedan afectar la apariencia de tu sitio.

Optimización del Rendimiento: Combina y minimiza tus archivos CSS para que se carguen más rápido. También, considera usar preprocesadores como SASS para tener un código más limpio y fácil de mantener.

Mantenimiento y Reutilización de Código: Define variables para cosas como colores y fuentes, y organiza tu código en módulos que puedas reutilizar fácilmente en diferentes partes de tu proyecto.

Seguridad: Ten cuidado con los estilos que permites, evitando cualquier inyección de código peligrosa y controlando el acceso a tus hojas de estilo.

JavaScript:

Organización del Código y Nombres Claros: Dale nombres descriptivos a tus variables y funciones. Organiza tu código en funciones pequeñas y reutilizables para mantenerlo limpio y fácil de seguir.

Validación y Pruebas: Escribe pruebas para verificar que tu código funciona como esperas y usa console.log para depurar durante el desarrollo. Recuerda eliminar estos mensajes antes de lanzar tu proyecto.

Optimización del Rendimiento: Evita usar bibliotecas o frameworks grandes si no son necesarios, y minimiza tus archivos JavaScript para mejorar la velocidad de carga de tu sitio.

Mantenimiento y Reutilización de Código: Divide tu código en módulos y funciones que puedas reutilizar. Utiliza herramientas como npm para gestionar las dependencias de tu proyecto.

Seguridad: No uses métodos inseguros como eval(), y asegúrate de que tu código esté protegido contra ataques como el Cross-Site Scripting (XSS).

PHP:

Organización del Código y Nombres Claros: Usa nombres claros para tus funciones y variables, y organiza tu código en archivos y carpetas según lo que hace cada parte. Considera usar patrones como MVC para una mejor estructura.

Validación y Pruebas: Valida y limpia los datos que ingresan los usuarios para proteger tu sitio de problemas como inyecciones SQL. Usa PHPUnit para hacer pruebas unitarias y asegurarte de que todo funcione correctamente.

Optimización del Rendimiento: Usa técnicas de caché, como Opcache, para hacer que tu sitio web sea más rápido, y optimiza las consultas SQL para evitar sobrecargar tu base de datos.

Mantenimiento y Reutilización de Código: Divide tu código en funciones y clases que puedas reutilizar, y considera usar un framework PHP como Laravel para facilitar la organización y el desarrollo.

Seguridad: Asegúrate de escapar todas las entradas del usuario y usa sentencias preparadas para tus consultas SQL. Protege tus archivos y directorios sensibles para evitar accesos no autorizados.