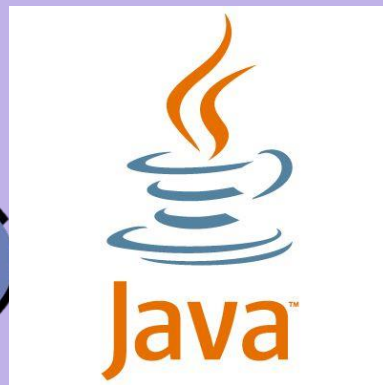


SUBMODULO: Desarrolla Aplicaciones con
Conexion a Base de Datos

Mtro. Gabriel Ignacio China Cortez

Actividad 1



ALUM. Flores Estrada Katya Naomi

Semestre 5 – Programacion

Jueves 29 agosto 2024

FUNCIONAMIENTO DEL PROGRAMA

Formulario

file:///D:/5º Modulo China/Act1 Formulario/index.html

FORMULARIO ACTIVIDAD 1

Nombre(s):

Apellidos:

Telefono:

Correo:

Alias:

Búsqueda

ESP LAA 11:26 p. m. 29/08/2024

Formulario

file:///D:/5º Modulo China/Act1 Formulario/index.html

FORMULARIO ACTIVIDAD 1

Nombre(s):

Apellidos:

Telefono:

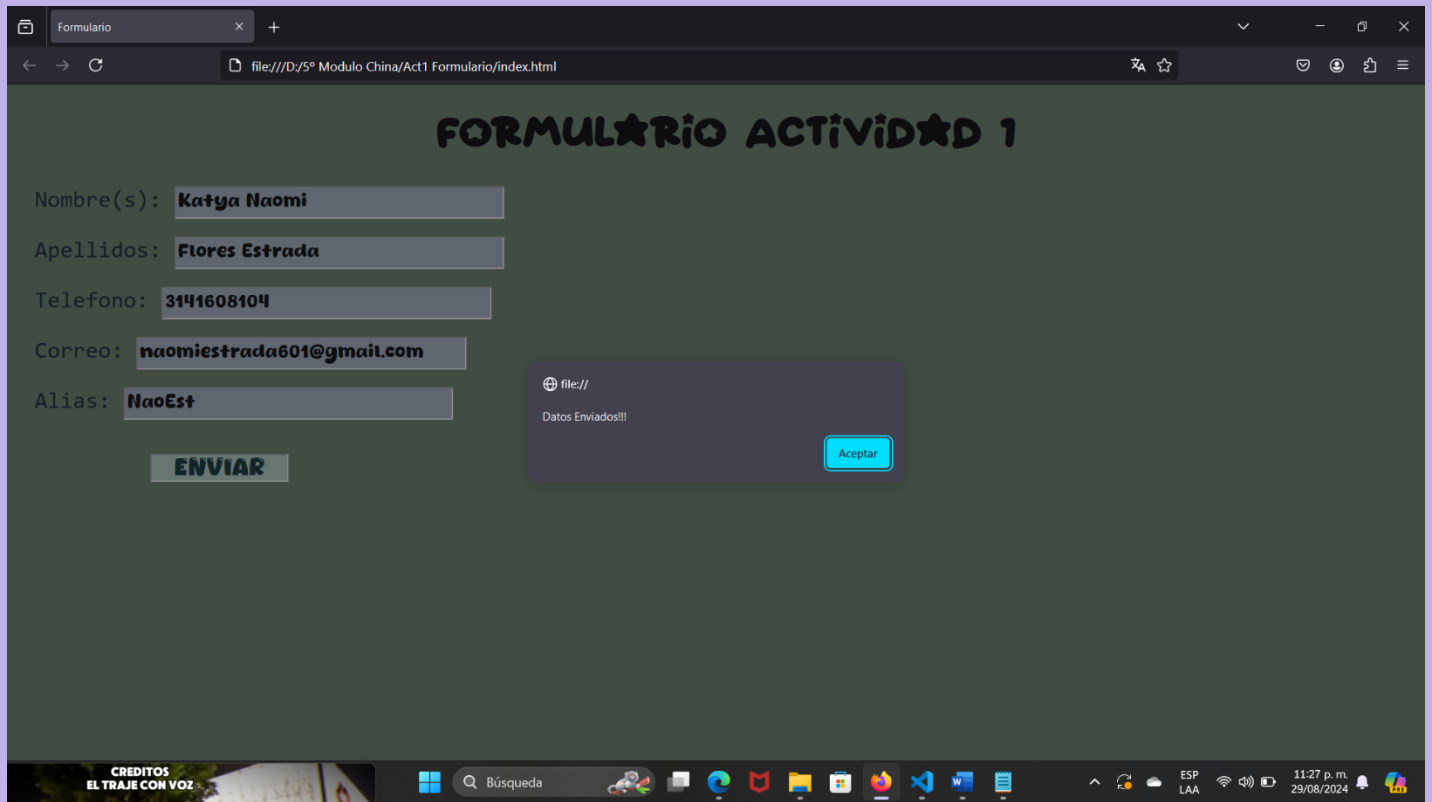
Correo:

Alias:

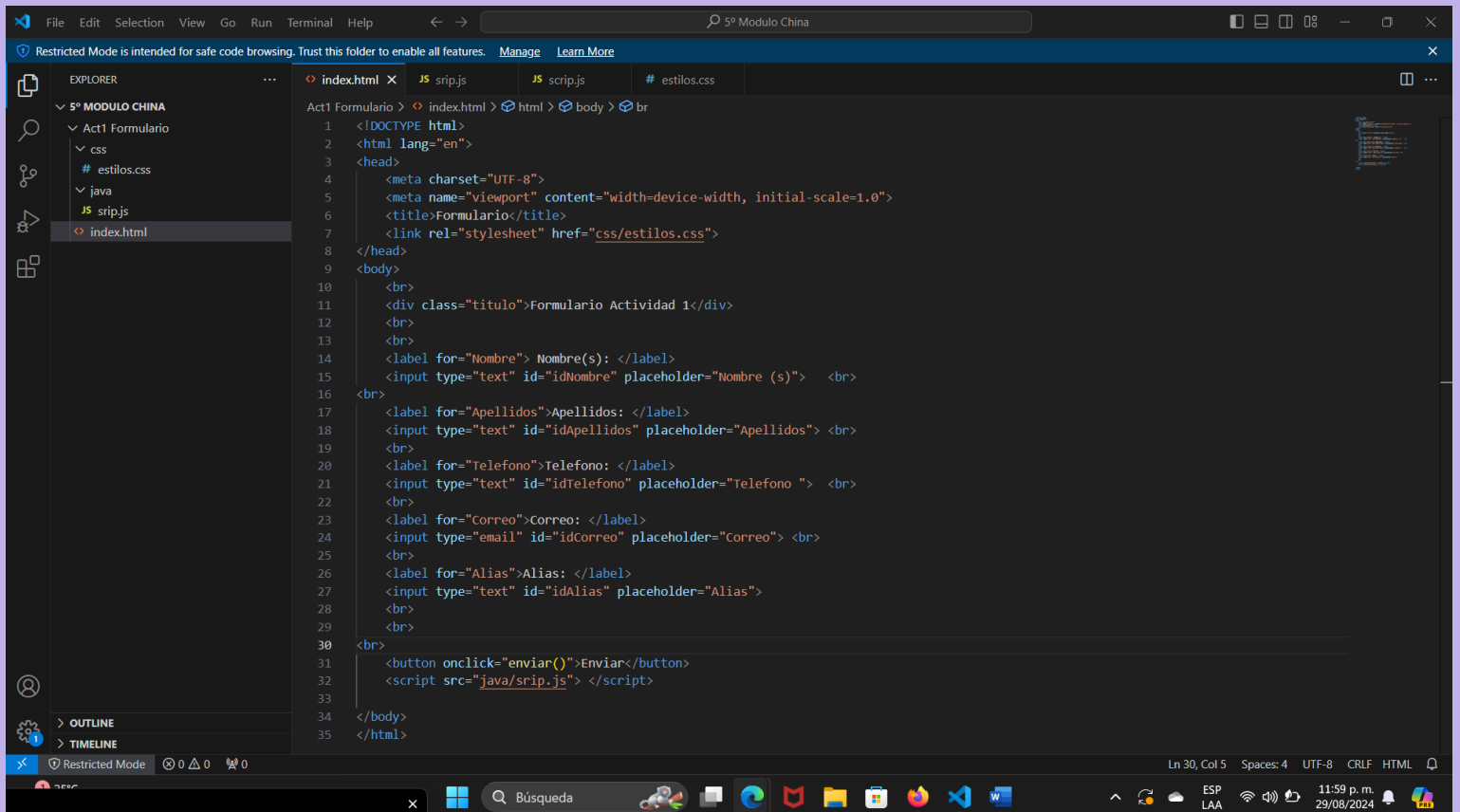
CREDITOS EL TRAJE CON VOZ

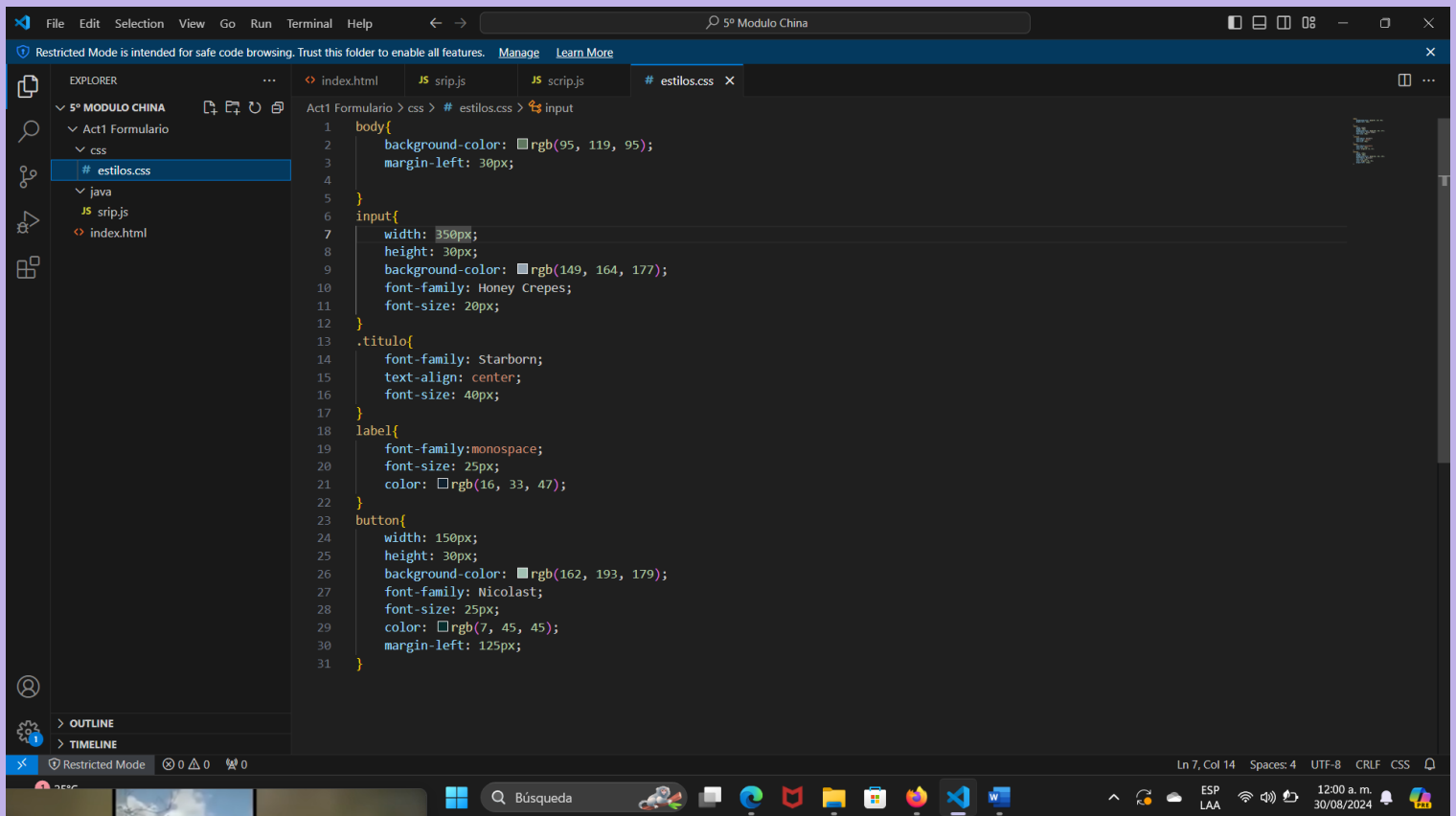
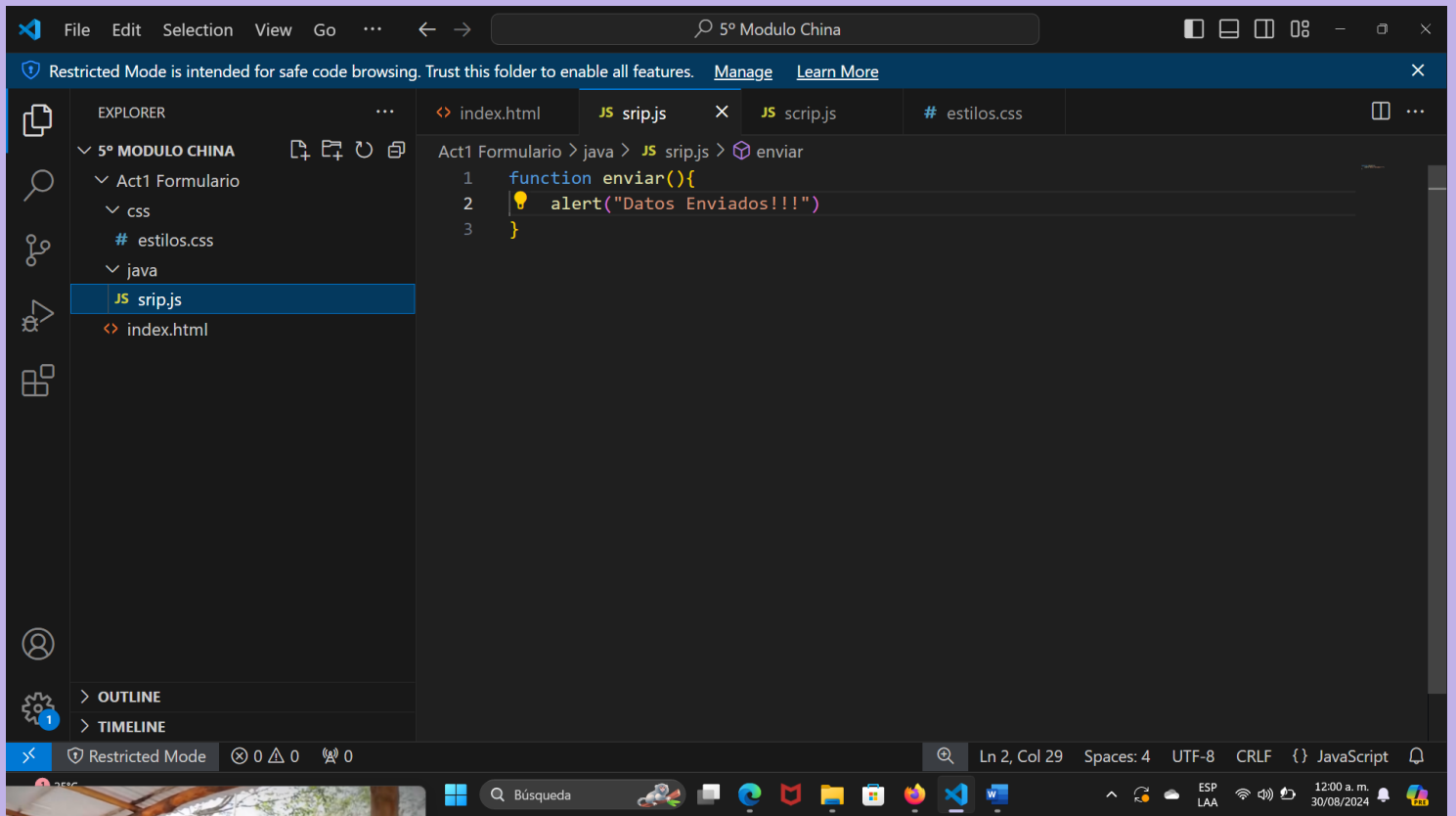
Búsqueda

ESP LAA 11:27 p. m. 29/08/2024



IMÁGENES DEL CODIGO HTML, CSS,





Buenas Practicas

HTML

- ☐ Utiliza etiquetas como `<header>`, `<footer>`, `<article>`, `<section>`, etc., para estructurar tu documento de manera clara y significativa. Esto mejora la accesibilidad y SEO.

- ☐ Usa un validador de HTML como el de W3C para asegurarte de que tu código sigue los estándares y no contiene errores.

- ☐ Siempre proporciona descripciones alternativas (`alt`) para las imágenes. Esto es esencial para la accesibilidad y el SEO.

- ☐ Mantén una estructura clara con etiquetas correctamente anidadas. Evita errores como cerrar etiquetas en el orden incorrecto.

Añade comentarios para explicar secciones complejas del código, facilitando su mantenimiento y comprensión.

CSS

- ☐ Mantén los estilos en archivos CSS separados para mejorar la mantenibilidad y la reutilización.

- ☐ Utiliza preprocesadores como SASS o LESS para escribir CSS más limpio y modular.

- ☐ Usa media queries y unidades relativas (`em`, `%`, `vh`, `vw`) para asegurar que tu diseño se adapta a diferentes dispositivos.

- ☐ Para mejorar el rendimiento, combina y minimiza los archivos CSS para reducir el tiempo de carga.

- ☐ Emplea una metodología como BEM (Block, Element, Modifier) para mantener el código CSS organizado y fácil de leer.

JAVA

- ☐ Adopta los principios SOLID para diseñar sistemas orientados a objetos que sean robustos y fáciles de mantener.
- ☐ Siempre usa un sistema de control de versiones como Git para mantener un historial claro y organizado del código.
- ☐ Utiliza comentarios y javadoc para documentar tus métodos y clases, facilitando la comprensión y mantenimiento del código.
- ☐ Maneja las excepciones de manera adecuada, evitando el uso excesivo de bloques try-catch y usando excepciones personalizadas cuando sea necesario.
- ☐ Utiliza estructuras de datos y algoritmos apropiados para mejorar el rendimiento de tu aplicación. Además, evita el uso innecesario de recursos.

PHP

- ☐ **Utiliza nombres de variables y funciones descriptivos**
- ☐ **Organiza el código en funciones y clases reutilizables**
- ☐ **Utiliza comentarios para explicar el código complejo**
- ☐ **Asegúrate de que el código sea seguro y validado**
 - ☐ **Utiliza la documentación php para seguir las mejores prácticas.**