

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Introducción a la Informática

INFORME
Proyecto Fase II - IALI

Preparador:
Bruno Lugo

Estudiantes:
Victor Casillas (30.606.977)
Violeta Semprúm (31.423.116)

Caracas, julio de 2025.

Proyecto: Educonnect.

Extensiones: HTML, CSS, JS y Archivos JSON.

Archivos:

Css involucrado:

- style.css

Js involucrado:

- Script.js

HTML involucrados:

- CatalogoCurso.html: Vista con los cursos disponibles de EduConnect.
- ConfirmacionCurso.html: Vista con la información del curso y confirmación de inscripción.
- CursoDetalle.html: Vista con los detalles del curso, los materiales, el proyecto final, la posibilidad de cargar un archivo y, una vez subido, los detalles de la calificación.
- login.html: Vista para iniciar sesión.
- MisCursos.html: Vista de los cursos ya inscritos.
- perfil.html: Vista para consultar la cuenta.
- RegistroUsuario.html: Vista para registrar una cuenta.

Json involucrados:

- curso.json
- estudiantes.json

Metodología usada:

EduConnect es un proyecto que simula una página de cursos para estudiantes. La página principal de este es el catálogo que muestra los cursos disponibles a ofertar a los usuarios. Para poder acceder a ellos, el usuario debe iniciar sesión o registrarse en la página llenando unos formularios. Habiendo accedido, tiene la posibilidad de inscribir un curso, revisar los cursos ya inscritos y ver los detalles de los mismos, donde además puede revisar los materiales de uno, subir un archivo que simula la entrega de un proyecto final y recibir la calificación por el mismo. Como extra, se implementó la posibilidad de que el usuario pudiera consultar su perfil.

Manejo de Archivos JSON:

Se crearon dos archivos json con la información de las cuentas de unos estudiantes y la de los cursos a ofertar. Para leer el contenido de los mismos y mostrarlos en los html correspondientes, se hizo uso de la función `fetch()` en javascript. En el caso de los usuarios, se lee el contenido del json con la función `cargarestudiantes()` y este es transferido a un arreglo dinámico en `localStorage` llamado `estudiantes`. La función solo es ejecutada si dicho arreglo está vacío, para evitar que la información guardada durante la ejecución del proyecto sea eliminada.

En el caso de los cursos, se lee el contenido del json con la función `mostrarcursos()`, donde cada uno de los elementos y sus atributos son agregados al html `CatalogoCurso` en conjunto con un botón que logra identificarlos según el id. La función se ejecuta cada vez que se accede a la interfaz `CatalogoCurso`.

Manejo de memoria:

Para el manejo de memoria y pase de datos se hizo uso del `localStorage` con las siguientes variables.

- **'estudiantes'**: Para tener guardada la data de las cuentas, se creó este arreglo. Este recibe los objetos de `estudiantes.json` si está vacío y en el caso de que el usuario registre una cuenta nueva, el sistema crea una instancia de la clase `'Estudiante'` para que luego sea agregada al arreglo con el fin de guardarla. Este arreglo es accedido al iniciar sesión, registrar una cuenta y cuando un estudiante inscribe un nuevo curso, con el propósito de actualizar los cursos en los que su cuenta está registrada.
- **'loggedUser'**: Esta variable almacena la cuenta con la que la sesión fue iniciada o que se registró, es usada para identificar al usuario en lo que navega entre las interfaces y realiza distintas operaciones.
- **'misCursos'**: Es una variable que guarda el usuario de la cuenta loggeada y por cada uno almacena un arreglo con los ids de los cursos en los que está inscrito, de modo tal que al inscribir uno nuevo o revisarlos, el usuario pueda acceder a ellos.
- **'entregas'**: Para que el estudiante realice el proyecto final de un curso. Es una variable con las claves usuario, donde almacena el nombre de usuario de la cuenta loggeada, y curso, donde almacena el id del curso en el que el usuario está entregando su proyecto. Cuenta con campos donde se guarda el nombre del archivo subido y la calificación y los comentarios del profesor.

Entre las consideraciones que tiene el proyecto, agregamos que como el único actor presente en esta fase fue el estudiante, al éste subir un proyecto a un curso, se genera una calificación con un número random y se muestra un comentario según dicho número. Para ver estos datos tras la entrega, se debe volver a acceder manualmente al curso con el fin de poder desplegar la nueva sección de calificaciones.

Solo fue usado un archivo js para agregarle comportamiento a la página y un archivo css para agregarle los estilos.

El proyecto debe abrirse usando un servidor local como Live Server en VS Code, de esa manera el json será leído correctamente. Click derecho sobre CatalogoCurso.html y seleccionar "Open with Live Server". Asegurarse de que se está accediendo al proyecto desde las rutas: <http://localhost:5500/> o <http://127.0.0.1:5500/> , en caso de que no lo estén, pegarlas y entrar a la vista CatalogoCurso.html.