

Evidencia de competencia

A01709577 – Roberto Serna Niño

Competencia a evaluar:

STC0204 – Desarrollo de componentes de software

¿Qué desarrollé en este proyecto?

En este proyecto trabajé en el desarrollo del back-end y un poco del front-end de nuestra aplicación web. Trabajé sobre controladores y modelos dónde implementé diferentes funciones y métodos que nos sirvieron para trabajar las diferentes funcionalidades, asimismo, he tratado de ordenar por bloques el código y separar mis aportes para mantener la lógica del sistema y el orden de trabajo.

Esta estructura “modular” nos permitió implementar componentes del sistema de la mejor manera con la posibilidad de reciclar código o funciones que ya tenemos, cumpliendo con los lineamientos de la competencia.

¿Cómo apliqué buenas prácticas de codificación?

Durante el desarrollo las siguientes prácticas de codificación:

- Nombres de funciones con camelCase

```
static async findByEmail (
}

static async getRoleById (
}

static async getIdEmpresa
}

static async isCompanyAlr
}

static async registerComp
```

- Validación temprana de parámetros (validar antes de ejecutar acciones)

```
if (!nombre || !correo || !contraseña || !id_empresa) {  
  return false;  
}  
  
try {  
  const conn = await db();  
  // Comprobar si el correo ya existe  
  const exist = await conn.execute(  
    "SELECT * FROM Usuario WHERE correo_electronico = ?",  
    [correo]  
  );  
}
```

- Manejo de errores con try/catch

```
} catch (error) {  
  console.error("Error updating user:", error);  
  throw new Error("Error al actualizar el usuario");  
}
```

- Constantes para módulos importados

```
const express = require("express");  
const fileUpload = require("express-fileupload");  
const session = require("express-session");  
const bodyParser = require("body-parser");  
const cookieParser = require("cookie-parser");  
const path = require("path");
```

- Agrupación de métodos relacionados en clases (home.model.js)

Dividimos las funcionalidades del sistema en archivos de control, el controlador (home.controller.js) gestiona las operaciones del sistema, el modelo (home.model.js) se encarga de operaciones específicas para diversas entidades así como conectar con la database, las rutas ([home.routes.js](#)) gestiona que acciones del controlador se ejecutan al llamar ciertas rutas, así como la transmisión de datos al render de la página y por último index.js que se encarga de la lógica de la app web misma.

Esta organización permite facilitar el mantenimiento, la colaboración entre los miembros del equipo y la posibilidad de escalar el sistema.

¿Cómo se reflejan los diseños de componentes en el sistema?

Para el desarrollo del back-end, me basé en los diagramas que elaboramos previamente como parte de la planeación del proyecto (Estados y Secuencia), en estos definimos cómo organizaríamos los distintos módulos del sistema y cómo sería la interacción entre sí. En particular, seguí la estructura propuesta en dichos diagramas para separar claramente la lógica del sistema, el acceso a datos y las rutas de este. Esto permitió asegurar la coherencia del flujo de información y la correcta integración entre los distintos componentes del sistema, como el controlador de estudiantes, el módulo de requisitos y las notificaciones de alerta.