**Informe Técnico del Proyecto "Sistemas Web EHU"**

**Autores**

* **Leslie Sanchez**
* **Marta Zulaica**
* **Juan Pablo Zambrano**

**Link del repositorio en github: https://github.com/JuanZambrano2000/proyecto\_Sistemas\_web\_EHU.git**

**Introducción**

Este proyecto es una aplicación web diseñada para gestionar y visualizar habilidades (skills). El desarrollo fue realizado en equipo, siguiendo los requerimientos del enunciado proporcionado en clase, aunque este presentó, para nosotros varias ambigüedades que demandaron una interpretación colaborativa. La aplicación incluye funcionalidades como la visualización de habilidades, puntuaciones, tareas asociadas, y una sección de medallas.  
También, es posible consultar las tareas asociadas a cada habilidad y marcarlas en el caso de que se han completado.

El proyecto se desarrolló utilizando **Node.js** y **Express.js** como base del servidor y **EJS** como motor de plantillas. Además, se incorporaron métodos de scraping para obtener recursos externos como los iconos utilizados.

**Configuración y Arranque**

El servidor de la aplicación está configurado para ejecutarse en http://localhost:3000/. Se puede iniciar con los siguientes comandos:

* **npm run dev**: Inicia el servidor en modo desarrollo.
* **node app.js**: Inicia el servidor en modo estándar.

**Importante**: Si con algún comando marca error, intentar ejecutar: **npm i,** para asegurar que cuenten con todas las dependencias necesarias.

Nosotros decidimos instalar nodemon, ya que nos parece más práctico para no tumbar el servidor con cada cambio.

**Estructura del Proyecto**

**Archivos principales**

1. **app.js**  
   Configura el servidor Express, define las rutas principales y maneja errores.  
   Principales aspectos de configuración:
   * Motor de plantillas: **EJS**.
   * Directorios estáticos: /public para archivos CSS, imágenes, y JS.
     1. En este punto tuvimos varias complicaciones para enlazar correctamente las views con los archivos css y main.js, ya que también queríamos integrar lo aprendido en unas asignaturas y hacer uso de Bootstrap, lo cual fue beneficioso en algunos aspectos, pero complico el enlace con otros archivos css y js
   * Rutas principales: / (index) y /users.

En index, tenemos varias rutas que nos ayudan al funcionamiento y fluidez de nuestra página. Cada ruta esta asociada a una view

* + - * Raiz(/), es la página de inicio
      * Login(/login), lleva al inicio de sesión, el cual no esta completamente implementado
      * Skills(/skills), Muestra todas las habilidades disponibles en formato interactivo
      * Leaderboard (/leaderboard), Presenta una tabla o lista de medallas y sus requisitos.
      * Detalle de Habilidades (/skills/:id), muestra información detallada de una habilidad específica, identificada por su id

Las rutas que dependen de archivos externos (competencias.json y badges.json) leen sus contenidos en tiempo de ejecución. En estos archivos es donde guardamos toda la información desplegada en el sitio.

**Funcionalidades Implementadas**

**1. Visualización de habilidades (Skills)**

* Presenta las habilidades en un diseño hexagonal. Cada habilidad incluye:
  + Nombre.
  + Icono.
  + Puntuación (score).
  + Descripción.
  + Lista de tareas asociadas, con opción de marcar como completadas.

Los primeros dos puntos, fueron obtenidos con la función scraper. El resto fue agregado manualmente.

**2. Detalle de habilidades**

* Al seleccionar una habilidad, se muestra información detallada:
  + Tareas con checkboxes.
  + Animación de confetti al completar todas las tareas.
  + Sección de subida de evidencia (formulario), que se activa tras completar todas las tareas.

**3. Sistema de medallas (Leaderboard)**

* Página dedicada a la consulta de las medallas disponibles y sus requisitos.

**4. Inicio de sesión (Login)**

* Por el momento, la funcionalidad es limitada. El botón "Ingresar" redirige directamente a la página de habilidades
* Asi mismo, existe el botón de cerrar sesión en skills, sin embargo, no manejamos aun sesiones, por lo que solo redirige a la página de inicio.

**Desafíos y Limitaciones**

**Interpretación del enunciado**

* La ambigüedad en las especificaciones del proyecto requirió constantes reuniones de equipo para interpretar y decidir cómo proceder. Esto afectó la eficiencia en algunas etapas.

**Scraping de Iconos**

* Se utilizó un método de **scraping** para obtener iconos asociados a las habilidades. Aunque funcional, esta técnica genera demoras en el tiempo de respuesta, y puede mejorarse utilizando un sistema de almacenamiento en caché o una base de datos local para almacenar recursos obtenidos previamente.