UNIVERSIDAD PRIVADA DEL VALLE
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS INFORMÁTICOS
UNIVALLE – COCHABAMBA

PROYECTO DE SISTEMAS III

Gamificacion

DOCENTES:

Christian Montano Salvatierra

Gaston Silva Sanchez

Cochabamba noviembre de 2023

Gestión 2

MANUAL TECNICO

1. Introducción:

Este manual describe los pasos que necesita el cliente y personas que quieran replicar la instalación de la página web de “Gamificación”.

Es importante tener en cuenta que en el presente manual se hace mención a las especificaciones mínimas de hardware y software para la correcta instalación de la página web.

2. Descripción del proyecto:

Este proyecto consiste en el desarrollo de una página web que brindara a los estudiantes badges y puntos para así subir de rango en la pirámide de rangos de la página y para que los badges de las investigaciones y trabajos que realicen aporte a su curriculum.

Los tipos de usuario son: Administrador, Gestor y Estudiante.

Las vistas que se tienen en el proyecto varían según el tipo de usuario que este inicie sesión como por ejemplo el administrador al iniciar sesión contara con distintos apartados, el de gestor contara con menos apartados siendo la mayoría de visualización y por ultimo los estudiantes que al iniciar sesión podrán ver su perfil y que es lo que ganaron o en que rango se encuentran y otras opciones más.

3. Roles/Integrantes

Integrantes	Roles
Luciana Mayra Blanco Aranibar	Team Leader, developer, DB architec, Integration, GIT Master
Steven Gabriel Claros Tapia	Developer

4. Arquitectura del software: Explicación de la estructura y organización del software, incluyendo los componentes principales, las interacciones entre ellos y los patrones de diseño utilizados.

La estructura y organización del software es que se realizo en un proyecto de “React” utilizando “tailwindcss” para el frontend de elementos de algunas vistas, “vite” para una compilación más rápida. Como base de datos se utilizó “SqlServer” y para las peticiones a este se utilizo una “API” desarrollada en “ASP.NET”

- En la parte de las vistas de la pagina se estableció la lógica y validaciones.
- La Api es la encargada de enviar estos datos realizando el CRUD correspondiente en la base de datos

5. Requisitos del sistema

- **Requerimientos de Hardware (mínimo):**
  - **Procesador:** Gama baja o media.

- **Memoria RAM:** Al menos de 2GB de RAM.
- **Almacenamiento:** Al menos 500mb.
- **Conexión a Internet:** Se recomienda una conexión estable.
- **Tarjeta de red:** El equipo debe contar con una tarjeta de red.

- **Requerimientos de Software:**
  - **Sistema Operativo:** Windows, MacOS.

6. Instalación y configuración: Instrucciones detalladas sobre cómo instalar el software, configurar los componentes necesarios y establecer la conexión con otros sistemas o bases de datos

- Descargar los archivos del proyecto del GitHub.
- Instalar Node.js.

Una vez instalado estos se debe utilizar Visual Studio Code en la carpeta del proyecto y abrir una terminal para ejecutar los siguientes comandos:

- npm install

Una vez realizado ya se puede ejecutar el proyecto de manera local con los siguientes comandos

- npm run dev (en la carpeta del proyecto)

Para la parte de la API se tendría que revisar los puertos de conexión que se encuentra en cada página ya que estos son los encargados de enviar o recibir la informacion de la API (si es que estos cambian o exista algún problema de conexión).

Para la parte de la API se deben realizar cambios en la ruta de la base de datos como en el archivo de “appsetings”

```
  "ConnectionStrings": {
    "DefaultConnection":              "Server=(localdb)\\DESKTOP-5AJJOCD;Initial
Catalog=BDGamificacion;User
ID=sa;Password=Univalle;MultipleActiveResultSets=False;Encrypt=True;TrustServerC
ertificate=False;"

  },
```

También en la parte de bdgamificacionContext:

```
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
#warning To protect potentially sensitive information in your connection string,
you should move it out of source code. You can avoid scaffolding the connection
string by using the Name= syntax to read it from configuration - see
https://go.microsoft.com/fwlink/?linkid=2131148. For more guidance on storing
connection strings, see http://go.microsoft.com/fwlink/?LinkId=723263.
        => optionsBuilder.UseSqlServer("Data Source=DESKTOP-5AJJOCD ;Initial
Catalog=BDGamificacion          ;          user=sa;          password=Univalle
;Encrypt=False;TrustServerCertificate=False");
```

Después de realizar los cambios la aplicación debería de funcionar con normalidad.

7. GIT

- **versión final del proyecto React**

https://github.com/LucianaMayraBlancoAranibar/GAMIFICACION_III.git

- **versión final del proyecto API**

https://github.com/LucianaMayraBlancoAranibar/API_Gamificacion.git

8. Personalización y configuración: Información sobre cómo personalizar y configurar el software según las necesidades del usuario, incluyendo opciones de configuración, parámetros y variables.

Para configurar la parte de los colores de la pagina se debe revisar si se está usando css o tailwindcss ya que en caso de tailwindcss se debe cambiar desde la etiqueta mientras que en css se debe entrar a su archivo css.

9. Seguridad: Consideraciones de seguridad y recomendaciones para proteger el software y los datos, incluyendo permisos de acceso, autenticación y prácticas de seguridad recomendadas

1. Usa contraseñas seguras y almacenamiento seguro: Asegúrate de que las contraseñas de los usuarios sean lo suficientemente fuertes, utilizando combinaciones de letras mayúsculas y minúsculas, números y caracteres especiales. Nunca almacenes contraseñas en texto plano. En su lugar, utiliza funciones de hash seguras, como bcrypt, para almacenar y verificar las contraseñas.
2. Implementa autenticación segura: Utiliza técnicas de autenticación seguras, como tokens de sesión, cookies seguras y control de sesiones adecuado para verificar la identidad de los usuarios. Evita almacenar información sensible en las cookies y asegúrate de que las cookies estén marcadas como seguras y HttpOnly.
3. Limita los permisos de acceso: Asegúrate de que los usuarios solo tengan los permisos necesarios para acceder y modificar los datos. Establece niveles de acceso adecuados y utiliza roles de usuario para restringir el acceso a partes sensibles de la aplicación.
4. Protege contra ataques de fuerza bruta: Implementa medidas para proteger contra ataques de fuerza bruta en los formularios de inicio de sesión, como limitar el número de intentos de inicio de sesión y bloquear las direcciones IP después de múltiples intentos fallidos.
5. Utiliza HTTPS y certificados SSL: Implementa HTTPS en tu sitio web utilizando un certificado SSL válido. Esto cifrará la comunicación entre el cliente y el servidor, protegiendo los datos sensibles durante el transporte.

10. Depuración y solución de problemas: Instrucciones sobre cómo identificar y solucionar problemas comunes, mensajes de error y posibles conflictos con otros sistemas o componentes.

La depuración de problemas en la aplicación se da al ejecutar el proyecto ya que en la consola de visual studio code saldrá en que parte salió el error, también se podrán visualizar los errores de manera visual en la página del sistema web ya que estos no cargaran y se quedaran

blancos, además de que saldrá un mensaje en rojo el cual explicara el error, en caso de que los registros no se carguen en sus respectivas tablas revisar la consola del navegador web para ver si hay error en los llamados en la API.

11. Glosario de términos: Un glosario que incluya definiciones de términos técnicos y conceptos utilizados en el manual.

Certificados SSL: Son certificados de seguridad que se utilizan para establecer una conexión segura y cifrada entre el servidor web y el navegador del usuario. Estos certificados validan la identidad del servidor y garantizan que la comunicación sea segura y confiable.

Bcrypt: es una función de hash de contraseña que se utiliza para almacenar y verificar contraseñas de forma segura en aplicaciones web. A diferencia de algoritmos más simples, bcrypt utiliza una técnica de "salting" que agrega una cadena aleatoria adicional a la contraseña antes de aplicar el hash.

HttpOnly: es un atributo de seguridad que se puede configurar para las cookies de una aplicación web. Cuando una cookie tiene habilitada la opción HttpOnly, significa que solo puede ser accedida y modificada por el servidor a través del protocolo HTTP.

12. Referencias y recursos adicionales: ***Enlaces o referencias a otros recursos útiles, como documentación técnica relacionada, tutoriales o foros de soporte.***

https://es.react.dev

https://tailwindcss.com/docs/installation

13. Herramientas de Implementación:

- **React**
- **Vite**
- **css**
- **JavaScript**
- **Node.js**
- **Html**