**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 la “Sociedad Científica de Administración de Empresas” (SOCAE).

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.

1. **Descripción del proyecto:**

Este proyecto consiste en el desarrollo de una página web que proporciona información general de la **SOCAE** y los distintos apartados que componen a esta como ser:

* People development.
* Marketing.
* Innovación.
* Investigación Científica.

También incluye un apartado de administración de contenido de la página web principal antes mencionada.

Este proyecto tiene la finalidad de compartir esta información con los estudiantes de la Universidad del Valle y permitir que los directivos de la SOCAE puedan gestionar la información que quieran compartir.

1. **Roles / integrantes**

|  |  |
| --- | --- |
| Integrantes | Roles |
| Luis Miguel Lovera Caro | Team Leader, developer, DB architec, Integration |
| Steven Gabriel Claros Tapia | Git Master, developer, QA tester, Integration |

1. **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 la organización del software que se utilizo es la de MVC usando la parte de los modelos, vistas y controladores.

* En la parte del modelo se estableció la lógica y las reglas de la página web donde se procesarán los datos que se envíen y se realizarán las validaciones.
* En la parte de la vista se encargará de la interfaz de usuario y la interacción con los usuarios, donde se tendrán los archivos PHP, SCSS entre otros.
* En la parte del controlador se encargará de la interacción con la base de datos donde se realizarán los CRUD correspondientes a sus apartados.

La interacción entre los componentes se utilizó el patrón de Diseño MVC: Como mencionado anteriormente, el patrón MVC se utiliza para separar la lógica de presentación, lógica de negocio y acceso a datos en componentes separados, lo que permite una mayor modularidad y facilidad de mantenimiento.

1. **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.
     + **Servidor web:** Apache.
2. **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 la versión 8.0.28 de PHP.
* Instalar Node.js.
* Instalar apache.

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 (dependencias del package.json)
* composer install (dependencias del php)

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

* npm run dev (en la carpeta del proyecto)
* cd public (para movernos a la carpeta)
  + php -S localhost:3000 (configurar el archivo .env antes con la base de datos)
* DB\_HOST=localhost
* DB\_USER=root
* DB\_PASS=(contraseña de base de datos)
* DB\_NAME=(nombre de la bdd)
* EMAIL\_HOST=sandbox.smtp.mailtrap.io
* EMAIL\_PORT=2525
* EMAIL\_USER=(de tu cuenta de mailtrap)
* EMAIL\_PASS=(de tu cuenta de mailtrap)
* HOST=https://localhost:3000

Esta sería la configuración del archivo .env para ejecutarlo de manera local.

1. **PROCEDIMIENTO DE HOSTEADO / HOSTING (configuración)**

* **Sitio Web.**
  + Para este caso utilizamos hostinguer donde se subieron los archivos del proyecto.
* **B.D.**
  + En la misma pagina del hostinguer tenemos la opción de usar la base de phpMyAdmin en la cual se subiría el script de la base de datos y lo siguiente seria cambiar el archivo .env para que se conecte a este.
* **Script de la bdd:**

** (**doble click para abrir)

* **.env**
* DB\_HOST=localhost
* DB\_USER=tu configuración de hostinguer
* DB\_PASS=tu configuración de hostinguer
* DB\_NAME=tu configuración de hostinguer
* EMAIL\_HOST=sandbox.smtp.mailtrap.io
* EMAIL\_PORT=2525
* EMAIL\_USER=(de tu cuenta de mailtrap)
* EMAIL\_PASS=(de tu cuenta de mailtrap)
* HOST=https://tudominiodehostinguer

1. **GIT :**

* **Versión final entregada del proyecto.**

[**https://github.com/StevenGCT/socae-landing-page.git**](https://github.com/StevenGCT/socae-landing-page.git)

1. **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 contenedores, tags, tamaños de pantalla, etc se debe entrar al archivo \_mixins.scss.

Para la parte de colores y variables globales se debe entrar al archivo de \_varaibles.scss.

1. **Consideraciones de seguridad:**
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.
7. **Depuración y solución de problemas:**

En el caso de tener problemas con la página web incluimos una función llamada “debuguear” que se encuentra en la siguiente ruta includes/funciones.php esta nos permite verificar información que enviamos o que contiene las diferentes variables que utilizamos en nuestros controladores para hacer peticiones y demás.

1. **Glosario de términos:**

**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.

1. **Referencias y recursos adicionales:**

* [**https://sass-lang.com/documentation/**](https://sass-lang.com/documentation/)
* [**https://www.php.net/docs.php**](https://www.php.net/docs.php)

1. **Herramientas de Implementación:**

* **PHP**
* **MySQL**
* **SCSS**
* **JavaScript**
* **Node.js**
* **Composer**