Universidad Nacional Mayor De San Marcos

Facultad de Ingeniería de Sistemas e Informática

Reporte Del Desarrollo de Software Versión 2.0

Proyecto SCSE “SeMas”

Grupo 1

Callán Pérez, Sergio Antonio

Félix Huayhua, Axel Patrick

Huamán Ortiz, Emerson Raúl

Lipa Palacios, Renzo Alejandro

Pérez Bazán, Eysen Christopher

Sanchez Saldaña, Sergio Antonio

Villanueva Chirito, Mariano Alonso

Yataco Quispe, Jorge Armando

**Introducción**

En el presente reporte se detallan las características del software desarrollado, así como también el funcionamiento esperado realizado en el transcurso del desarrollo en cuestión. Asimismo, se mencionará el lenguaje utilizado tanto para el programa como para la base de datos.

**Desarrollo**

**Hito 1**

Al encontrarnos en las fases tempranas del desarrollo, el software solo cuenta con las funcionalidades de registro y el inicio de sesión (login), lo que permite el almacenamiento de los datos de los usuarios y el ingreso a la aplicación web de manera adecuada. Para la creación de estos módulos, se utilizaron las herramientas: HTML, CSS, JavaScript, PostgreSQL y, para la conexión local, NodeJS.

**Archivos de extension HTML**

Los dos archivos HTML creados para el proyecto actualmente están destinados para generar la interfaz gráfica con la que se relaciona el usuario, y con campos de entrada para colocar la información vinculada al registro y el login, respectivamente. En ambos casos se han empleado clases para abarcar los espacios necesarios en los inputs de manera entendible y para la vinculación con los archivos de tipo CSS, se hizo uso botones tipo submit para el funcionamiento de las validaciones, además de botones tipo onclick para la navegación entre ambas.

**CSS (Hoja de estilo en cascada)**

Por parte del código CSS, los archivos HTML llaman a este archivo llamado SCSE-ELR para definir el estilo general de sus páginas y componentes: al ser parte del ingreso inicial de la página no se requiere que el diseño sea distinto, por lo que se comparten características de fuente y fondo.

**JavaScript**

Los archivos JS utilizados marcan el funcionamiento del programa en sí. Tanto autenticar.js como conexión.js sirven para definir las funciones a utilizar por el programa principal: app.js.  Este último será el que se ejecutará con la herramienta NodeJS para utilizar un puerto para el servidor local y poder utilizar las funciones ya mencionadas.

**PostgreSQL**

La base de datos que se está utilizando en el proyecto se basa en PostgreSQL (14) y se utiliza gracias a NodeJS (versión 12.13.0) que permite utilizar directamente javascript en el lado del servidor. En cuanto al funcionamiento del guardado de datos y su comparativa, se han utilizado dos funciones que permiten limitar y permitir tanto ingreso como registro. Por un lado se encuentra la Función ingresarSistema, el cual se encarga de verificar que los datos ya hayan sido colocados con anterioridad en la base de datos y, de encontrarse, permitir el ingreso. Por otro lado, se encuentra registrarUsuario, que guarda los campos requeridos y consistencia el nombre de usuario para que no pueda repetirse.

**Hito 2**

Para el desarrollo del hito 2 en cuanto al desarrollo de software se agregó el módulo referente a los profesores y la creación de un perfil en el que se detallan los puntos necesarios para poder generar lo necesario para el desarrollo del hito 3.  Los módulos, al igual que en el anterior hito, han sido realizados con los mismos lenguajes de programación y programas necesarios para su funcionamiento al ser un desarrollo de tipo incremental. Además, se ha colocado una imagen base como logo representativo de la aplicación web.

**Archivos de extension HTML**

Se han agregado archivos de extensión HTML relacionados con la opción de ser profesor y el poder registrar un curso para enseñar a alumnos, eligiendo carrera y horarios para su posterior guardado en su apartado de la base de datos. Asimismo, se agregó una opción en la página para poder tanto cerrar sesión, el cual retrocede a la pantalla de inicio de sesión, y también para cambiar de versión de profesor a alumno y viceversa.

Para concluir, se tiene planeado desarrollar páginas por curso para la parte de usuarios y que se observen en estos las posibles clases: actualmente solo se ha creado uno del curso Aritmética como referencia, pues pertenece al siguiente hito y no a este.

**CSS (Hoja de estilo en cascada)**

A diferencia de lo realizado en el hito 1, en esta parte del desarrollo no se ha utilizado demasiado archivos de extensión CSS, puesto que se ha utilizado características de HTML para el diseño base de la página principal. Solo se ha utilizado un archivo CSS para el Menú de Profesores y definir una selección de tipo dropdown.

**Base de Datos**

Para el avance del hito 2, se ha creado la tabla profesor y alumno, se ha agregado 8 registros de la tabla profesor con el fin de realizar pruebas en el programa, además se ha creado un registro de alumno con el fin de realizar pruebas. También, se ha creado un campo más, llamado tipo en la tabla usuario, esto es con la finalidad de identificar el tipo de usuario, si es alumno o profesor, ya que cada uno tendrá un diferente entorno de interacción con la aplicación.

Por último, se ha modificado las funciones ingresar\_sistema y registrar\_usuario debido al cambio de la tabla usuario del campo tipo. Además, se creado las funciones:

* consultar\_cursos, esta función permite obtener los datos del anuncio del profesor por medio del area que va enseñar,
* consultar\_datos\_usuario, esta función permite obtener los datos del usuario que ingreso al sistema
* consultar\_profesor, esta función permite obtener los datos del profesor y el curso que enseña por medio del identificador de la tabla profesor.

**Javascript**

Para el avance del hito 2, se ha creado la interacción de los botones de los cursos que se ofrecen, la interacción para ver el anuncio de cada profesor, la creación de las tablas donde se muestran los anuncios por áreas, el método ajax, en javascript, para obtener los datos de la base de datos y el uso de la librería jquery para facilitar el desarrollo de la acciones de los botones.

**Node.js**

Para el desarrollo del lado del servidor para la entrega del hito 2, se ha creado los archivos usuarios.js y profesor.js, estos archivos nos permite llamar a la base de datos, el cual devolverá la consulta que se desea realizar. En el archivo app.js se han creado las variables globales y 4 métodos post para el llamado del servidor al cliente. Estos son:

* app.post usuarioLogueado, permite obtener los datos del usuario.
* app.post listadoCursosProfesor, permite obtener los datos de los anuncios del profesor por medio de curso.
* app.post consultarProfesor, permite obtener los datos del anuncio con respecto a un profesor seleccionado.
* app.post obtenerIdProfesor, solo permite obtener el id del profesor seleccionado.

**Hito 3**

Para el desarrollo del último hito se han agregado funcionalidades que dan forma al proyecto como tal y el funcionamiento general del mismo. En este, se han actualizado campos de la base de datos, se han pulido detalles del estilo de la página, actualizado según lo necesario los archivos que utilizan node.js y se ha subido el proyecto a la nube mediante el servicio gratuito render.com.

**Base de datos**

En la base de datos del proyecto se han creado tablas referentes a los comentarios que se le dejan al profesor, así como también el apartado de pagos y los mensajes de comunicación directa con este último. Asimismo, se han agregado funcionalidades ligadas a dichas tablas, las cuales son los siguientes:

* Función calificar\_profesor
* Función ActualizarProfesor
* Función actualizar\_curso\_profesor
* Función ingresar\_comentario
* Función consultar\_comentarios\_profesor
* Función ingresar\_mensaje
* Función consultar\_mensaje\_profesor
* Función ingresar\_mensaje\_privado
* Función pagar\_profesor

**Node.js**

De cara al backend realizado con node.js, se actualizó el archivo profesor.js para que se ajuste a las necesidades actuales del proyecto, así como también el archivo base de la aplicación app.js. Además, para utilizar los nuevos campos de la base de datos, se han creado archivos .js para el pago, los mensajes y los comentarios anteriormente mencionados.

**HTML**

Los cambios en los HTML han radicado principalmente en acomodarse a las nuevas interfaces que se han implementado en cuanto a estilo, siendo los afectados el Buscador, Login, Menuprofesor, Register y Profesor, además de haberse creado un nuevo html nombrado pago.

**Estilos**

Para el apartado de estilos, se han utilizado dos tipos de archivo principalmente: las imágenes, siendo casos como el logo, profesor y backgrounds, así como también 2 nuevos css: SCSE-MP y SCSE-MP2.

**Conclusiones**

Para la parte final del proyecto se ha optado por pulir ciertas asperezas y agregar funcionalidades principales para poder mostrar el funcionamiento del proyecto y el estilo visual que se pensaba seguir de acuerdo a la coordinación interna del equipo mediante Figma.