**Prodoc - Descripción de componentes**

Jose David Giraldo / Arturo Gomez / Camilo José Montoya

Programación web

### Componentes de las páginas:

* **Login:** En esta página se renderiza un texto de introducción a la aplicación, un formulario con el correo del usuario, un botón para continuar y un link para registrarse.
* **Confirm ar login:** Pantalla a continuación del login de donde se confirmar el correo ingresado, si este no está registrado se llevará a la pantalla de registro y si lo está se le renderizara la foto de perfil y un formulario para la contraseña y un botón de ingresar.
* **Registro:** En esta pantalla se encuentra un formulario con los campos para llenar: nombre, correo, contraseña, teléfono y foto de perfil. Y un botón para registrarse.
* **Proyectos:** En esta página se renderiza una header con un texto de título “Proyectos” y abajo la cantidad que se tenga actualmente, a la derecha del header se encontrará un botón para filtrar proyectos por un tag. En adición, el contenido que sería un thumbnail de cada proyecto con una imagen del proyecto, un título, unos tags con los que está relacionado, la fecha de creación y una lista de los usuarios que están dentro de este. En la parte inferior, estará un navbar con tres botones, uno para el menú, otro para crear proyecto y otro para las n otificaciones.
* **Menú:** Similar a proyectos, ya que tiene el mismo navbar inferior, pero dentro de está se renderizara adicionalmente se mostrará una foto de perfil con el nombre y el correo para editarlo, y un botón para cerrar sesión.
* **Notificaciones:** En esta pantalla se muestran todos los cambios hechos en los proyectos que sean por adición o eliminación de archivos, asignación o finalización de tareas y el aviso para revisar el chat.
* **Tutorial:** Pantalla que se mostrará al crear el primer proyecto donde se explicará en unos cuantos pasos el cómo utilizar la interfaz. Por lo tanto, contiene textos, imágenes y el botón para continuar cuando el usuario termine de leer la información. Al final el botón tendrá la opción de crear proyecto para utilizar lo aprendido en el tutorial.
* **Crear proyecto:** En esta página se renderiza un texto de acompañamiento y un formulario para el nombre del proyecto y otro para agregar integrantes del proyecto mediante su correo, un botón para agregarle foto de perfil al proyecto, un input para los tags del proyecto y el botón para crear el proyecto y pasar a la página proyectos.
* **Chat:** Pantalla en la que se mostrará todo lo conversado en el proyecto y los elementos como eventos, archivos, links o tareas que se agregaron recientemente. También se mostrará cuando una tarea a sido completada por el usuario designado.
* **Agregar elemento:** En esta pantalla se tendrá un texto explicativo y un botón que desplegará opciones como: agregar archivo, link, evento o tarea.

### Vista:

**user:** Variable que guarda toda la información del usuario logueado

**projects:** Variable que guarda todos los proyectos del usuario logueado

**createInit:** Función que se encarga de crear toda la interfaz de la pantalla principal donde el usuario hace login. Carga los textos, imágenes, formularios y botones de estos respectivamente

**createRegister:** Función que carga el formulario necesario para registrar al usuario y el botón con el cual envía sus datos para el registro

**createHome:** Esta función renderiza el home con todos los proyectos, y las opciones de notificaciones y el menú

**getProject:** Función que crea un proyecto individualmente con toda la información que tiene en el momento

**getUser:** Esta función renderiza los usuarios al verlos dentro del proyecto individualmente

**getNotification:** Función que crea una notificación con el contenido que este posea, y se renderizara dependiendo del tipo de notificación que sea

**render:** Función que renderiza todos los elementos de la vista

### 

### 

### Eventos del controlador:

**refresh:** Este evento se llama al refrescar los proyectos que se tienen actualmente para saber si hay nuevos a los que el usuario esté relacionado.

**notify:** Evento para reconocer cambios cada cierto tiempo en la base de datos, y actualizar la información cambiada, solo si está relacionada con el usuario y sus proyectos.

**onLogin:** Evento que se ejecuta al hacer login y recibe las variables ***usuario*** y ***contraseña***para mandarlos al servidor y realizar la validación en nuestra colección de usuarios en la base de datos.

**onRegister:** Evento utilizado para registrar a un nuevo usuario, recibiendo las variables: ***nombre, correo, contraseña*** y ***foto de perfil***. Esta se envían al servidor y se guardan como nuevo usuario en la base de datos.

**onProject:** Este evento creará un nuevo proyecto en la base de datos recibiendo las variables ***titulo, integrantes, tags*** y ***preview***.

**onAddUser:** Agregara un nuevo integrante al proyecto seleccionado utilizando el objeto enviado: ***integrante***.

**onFilter:** Evento que filtra los proyectos del usuario por sus tags y retorna la nueva lista filtrada.

**onDelete:**Este evento eliminará la relación del proyecto con el usuario que esté logueado en el momento, si este es el dueño eliminará el proyecto por completo. Recibirá las variables: ***usuario*** y ***ident***.

**onChat:** Evento para enviar un nuevo mensaje para enviarselo a la base de datos recibiendo las variables: ***contenido, foto de perfil*** y ***nombre***.

**onFile:** Este evento enviará a la base de datos un nuevo archivo que se agregara al proyecto, puede ser imagen, audio o un documento. Recibirá los parámetros: ***file, ident*** y ***usuario***.

**onEvent:** Evento encargado de crear un nuevo evento en el proyecto como días de pre-entrega o presentación final, recibirá las variables: ***usuario, dia, hora, ident*** y ***titulo***.

**onTask:** Este evento crea una nueva tarea a un usuario específico, recibe los parámetros: ***nombre, delegado, titulo*** y ***fecha***.

**onLink:** Evento para compartir un link en el chat del proyecto requiriendo las variables: ***url, nombre, titulo*** y ***preview***.

### Modelo:

**API:**

* **post/api/login** *(correo y contraseña)***:** Revisa la existencia del usuario en la base de datos y retorna toda la información del usuario encontrado
* **post/api/register** *(nombre, correo, psw y foto de perfil)***:** Agrega al usuario en la base de datos con todos los campos brindados
* **get/api/:email/projects** *(correo)***:** Busca los proyectos relacionados con el usuario y retorna la lista resultante
* **get/api/:ident** *(identificación del proyecto)***:** Obtiene toda la información de un proyecto en específico
* **post/api/:ident/:email** *(correo)***:** Introduce a un nuevo usuario al proyecto
* **post/api/:email/:ident** *(correo e identificación del proyecto)***:** Introduce el nuevo proyecto relacionado con el usuario
* **post/api/:ident/linkList/:link** *(link)***:** Ingresa el nuevo link a la lista de urls de un proyecto específico
* **post/api/:ident/chat/:msg** *(mensaje)***:** Envía un nuevo mensaje a la lista de mensajes del chat
* **post/api/:ident/fileList/:file** *(archivo)***:** Agrega el archivo enviado a la lista de archivos de un proyecto en específico
* **post/api/deleteUser/:email** *(correo e identificación del proyecto)***:** Elimina al usuario de un proyecto específico y de su lista de proyectos