**Acta de constitución del proyecto Múuch' Wa'akun**

El patrón que se utilizo para el desarrollo del sistema fue MVC, específicamente con la tecnología MEAN

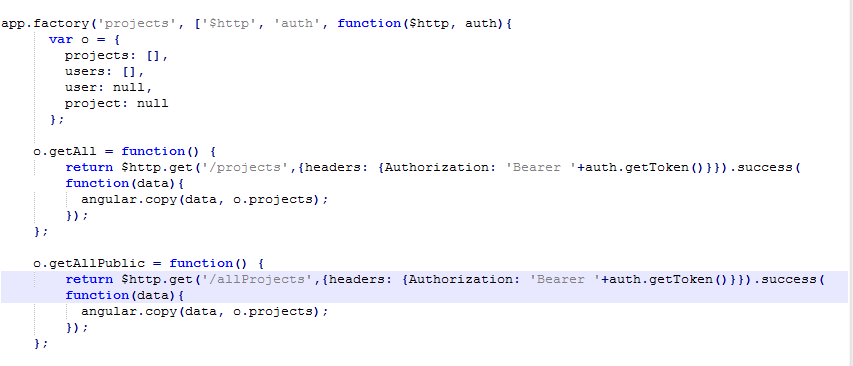
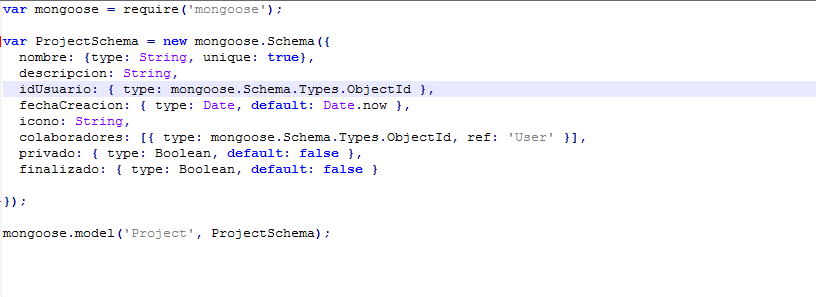
* MongoDB
* Express
* AngularJS
* NodeJS

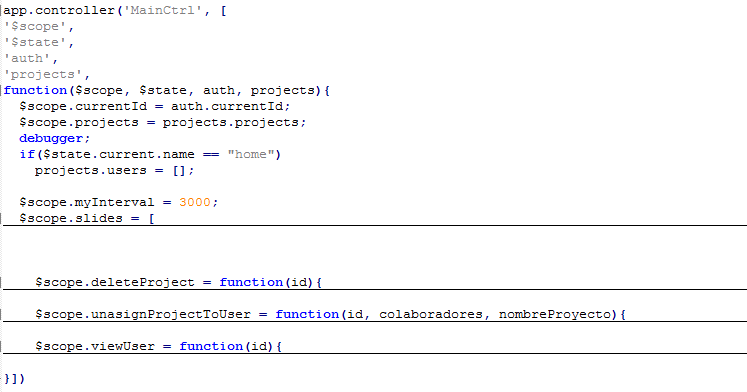
Se requiere instalar en el ordenador mongoDB y NodeJS, una vez instalados se instala por medio de línea de comandos Express y se genera un proyecto, de igual manera se instala mongoose en el proyecto para poder modelar y gestionar la base de datos con mongoDB.

En la parte del front end se genera una vista en la carpeta ‘views’ en dicho documento con extensión .ejs

Se especifica la estructura base del sitio y los diferentes templates para las secciones los cuales son llamados desde el controlador gestionado por angularJS el cual nos permite determinar estados y la carga de datos específicos por medio de métodos REST, para esto se genera un archivo javascript en la carpeta publica siendo referenciado desde nuestra vista, cabe recalcar que en la vista se deben agregar las referencias de angular.js y angular-routing desde cdn.  
  


Declaración de estado para una sección en particular

  
Ejemplo de factory para proyectos y funciones, entre ellas obtener los productos públicos haciendo una petición GET  
  
Declaración de rutas en archivo routes/index.js, entre ellas la ruta get para obtener proyectos públicos, en la cual se hace un query con mongoose para obtener los documentos tipo “Project” que cumplan con la condiciónal.  
  
Ejemplo de declaración de modelo para ser gestionado como documentos ligando la información a la base de datos mongo, se especifican las propiedades y tipos de datos del esquema, la gestión de las tablas es implícita por parte de mongoose.



Ejemplo de controlador el cual contiene funciones y propiedades asigandas al scope las cuales son consumidas desde la vista.



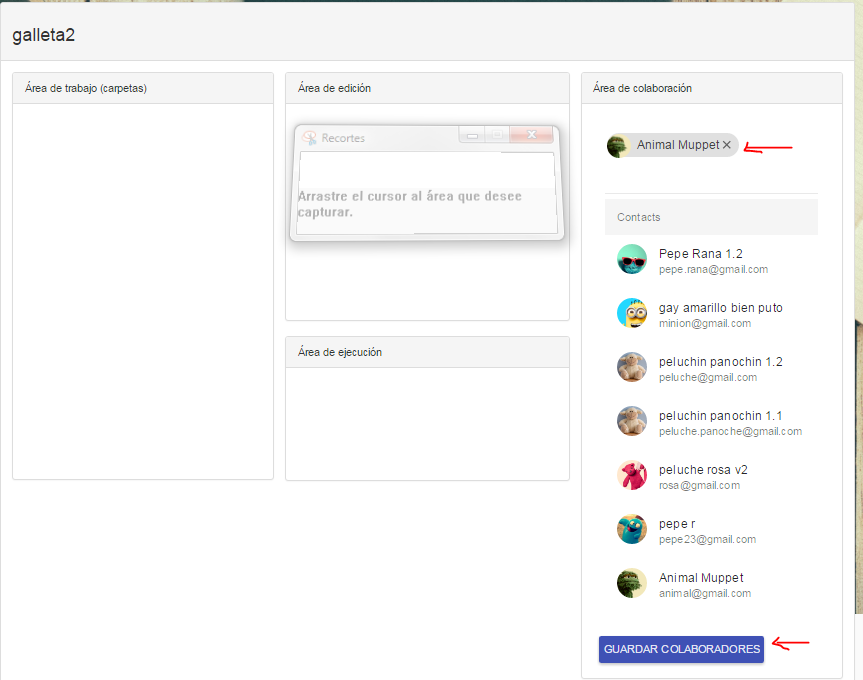
Ejemplo de template en la vista, el cual itera los proyectos de la variable “projects” del scope del controlador.

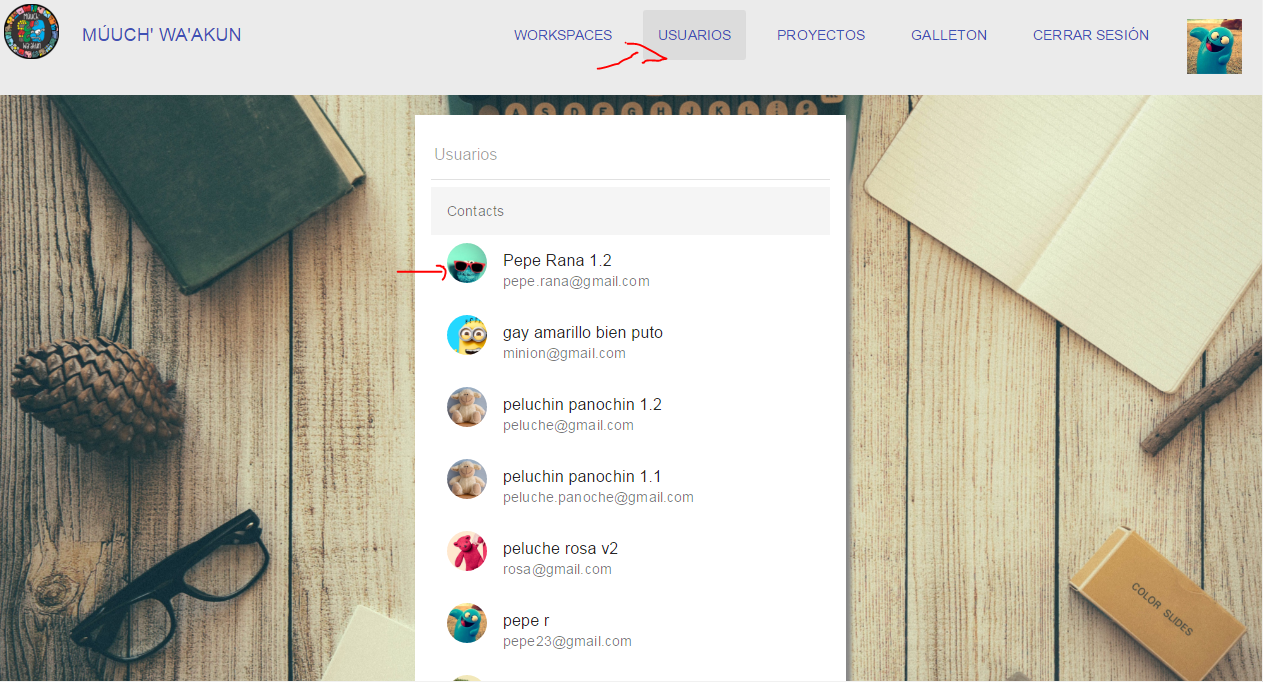
Los servicios REST son declarados en el index.js de la carpeta de ‘routes’ para lo cual es necesaria la referencia de mongoose y los modelos creados con esta tecnología, dichos modelos son creados en una carpeta ‘models’ y es ahí donde es especifica la estructura de los documentos a utilizar así como sus funciones.

Para el proyecto fueron necesarios el modelo User y Project los cuales gestionan en base de datos la información pertinente, siendo creadas las tablas y las actualizaciones de la información con ayuda del enlace de mongoose el cual lo hace ágil y transparente.  
  
Dichos modelos deben ser referenciados en el archivo de configuración app.config localizado en raíz de nuestro sistema, así como la referencia a Passport, librería utilizada para el flujo de autenticación de usuarios la cual también es instalada vía consola de comandos por medio de las instrucciones npm.  
  
El flujo de interfaces para el sistema consta de la pantalla de login y registro de las cuales inicia el flujo y ya siendo autentificado el usuario es redirigido a la pantalla de inicio en la cual se muestra una galería de imágenes y texto referentes a el objetivo del sistema.  
  
En la parte superior de navegación se encuentra el logo y el título del proyecto los cuales tienen link a la página de inicio y del lado derecho se encuentran los botones de ‘proyectos’, ‘perfil de usuario’ y logout así como el avatar seleccionado.  
  
En la sección de proyectos se encuentra del lado derecho un listado de los proyectos existentes el cual consta del icono del proyecto seleccionado, nombre, descripción, fecha de creación así como los botones de acciones para abrir, editar y eliminar proyecto.  
  
Al accionar el botón de abrir proyecto somos redirigidos a la pantalla de detalle del proyecto en la cual se encuentran el área de trabajo, de edición, ejecución y colaboradores en la cual se realizan las funcionalidades primordiales del sistema. Del lado derecho tenemos una implementación de contact chips con el listado de usuarios existentes en el sistema para su asignación como colaborador.  
  
En la pantalla de proyectos al accionar el botón de editar nos redirige a la pantalla de creación/edición de proyectos y nos muestra la info cargada de el proyecto en cuestión lista para su edición, siendo editada la información al guardar dichos datos son reflejados en base de datos y en el sistema mismo.  
  
Regresando a la pantalla de proyectos al accionar el botón de eliminar se elimina dicho proyecto de base de datos y del listado en cuestión.  
  
Del lado derecho del listado se encuentra el botón de crear proyecto el cual nos redirige a la pantalla de creación/edición de proyecto en la cual proporcionando los datos y guardando la info generamos un nuevo registro que es visualizado en la pantalla principal de proyectos enseguida de ser guardado.  
  
Retomamos el flujo del menú principal en el cual tenemos un botón con el nombre del usuario el cual nos redirige hacia la pantalla de perfil de usuario en la cual podemos editar la información del mismo siendo actualizada en base de datos y en el sistema siendo guardada.  
  
Y por ultimo tenemos el botón de logout el cual nos saca de sesión y nos redirige a la pantalla de login.  
  
Para la implementación de estilos fueron utilizadas las librerías de bootstrap y angular material las cuales son referenciadas en la vista principal.

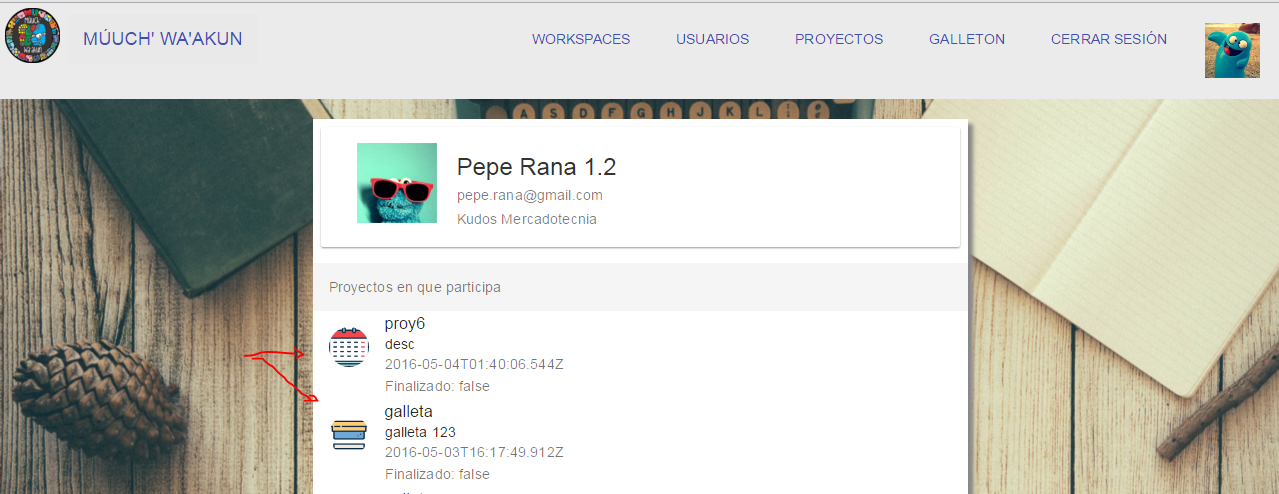
Anexo imágenes con respecto al flujo de pantallas:

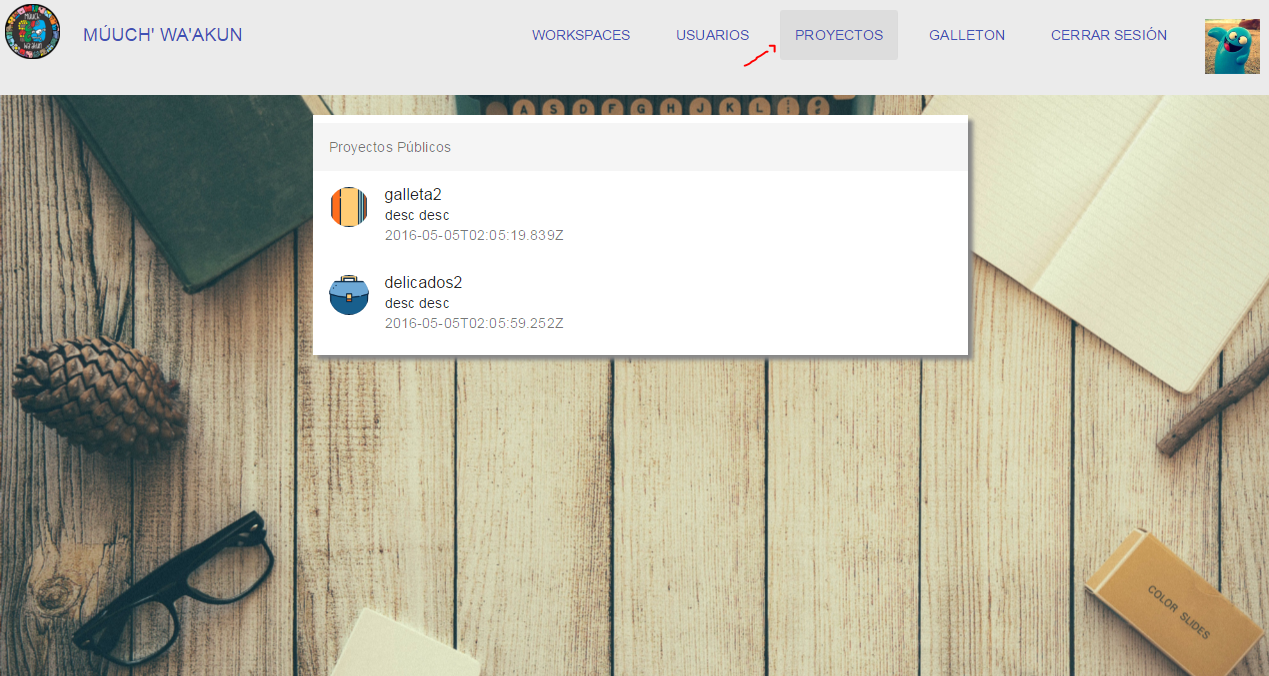


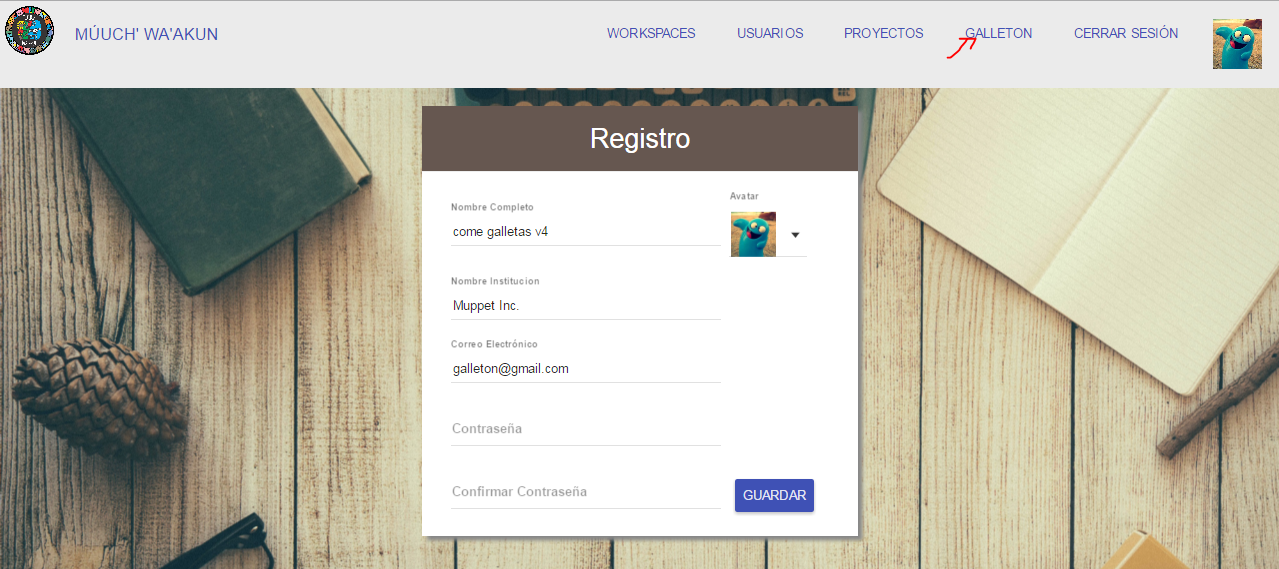
Guardar colaboradores de proyecto  


Usuarios en el sistema, para ver detalle click en la imagen  


Detalle del usuario con los el listado de los proyectos asignados



Proyectos en el sistema que son públicos  


Pantalla de edición de usuario logueado  


Perfil de usuario con información de proyectos asignados  
