

Tabla de Contenidos

- [Changelog](#)
 - [17/05/2021](#)
 - [18/05/2021](#)
 - [19/05/2021](#)
 - [20/05/2021](#)
 - [21/05/2021](#)
 - [21/05/2021](#)
 - [22/05/2021](#)
 - [26/05/2021](#)
 - [27/05/2021](#)
 - [28/05/2021](#)
 - [29/05/2021](#)
 - [30/05/2021](#)
 - **Finalizada la fase 1 (vease Fases del proyecto)**
 - [31/05/2021](#)
 - [1/06/2021](#)
 - [2/06/2021](#)
 - [3/06/2021](#)
 - [4/06/2021](#)
 - [5/06/2021](#)
 - [6/06/2021](#)
 - [7/06/2021](#)
 - [8/06/2021](#)
 - [9/06/2021](#)
 - [10/06/2021](#)
 - [11/06/2021](#)
 - **Finalizada la fase 2 (vease Fases del proyecto)**
 - [12/06/2021](#)
 - [12/06/2021](#)
 - [13/06/2021](#)
 - [14/06/2021](#)
 - [15/06/2021](#)
 - **Finalizada la fase 3 (vease Fases del proyecto)**
 - [16/06/2021](#)
 - [17/06/2021](#)
 - [18/06/2021](#)
 - [19/06/2021](#)
 - [20/06/2021](#)
 - [21/06/2021](#)
 - [22/06/2021](#)
 - [23/06/2021](#)
 - **Finalizada la fase 4 (vease Fases del proyecto)**
 - [Pendiente para mañana](#)
 - [Pendiente:](#)

- [Propuestas de mejora](#)

Changelog

17/05/2021

- Creacion de la estructura de paginas a seguir
- Creacion del modelo jerarquia de usuario
- Creacion del protocolo de seguridad

18/05/2021

- Creacion del sistema de login del usuario
 - Actualizacion del protocolo de login del backend
 - Declaracion de uso del sistema jwt para protocolo de seguridad
 - Declaracion del uso de REST para comunicaciones entre BackEnd y FrontEnd

19/05/2021

- Correcciones menores al sistema de login del FrontEnd
- Correcciones menores al sistema de login del BackEnd
- Agregado el sistema de roles para el Backend

20/05/2021

- Correcciones menores al NavBar
- Se establece un sistema para determinar el estado de la sesion del usuario mediante props
- Establecido sistema de login y soporte heredado navbar

Componente	Envia	Recibe
Pagina*	ServerloggedIn,token	
Layout	ServerloggedIn,token	ServerloggedIn,token
NavBar	ServerloggedIn,token	

* Cualquier pagina que llame a los componentes hijos a continuacion

- El menu de usuario cambia segun el tipo de usuario

Usuario Visitante

- Sobre Nosotros
- Ofertas
- Quienes somos
- Iniciar Sesion

Usuario Normal

- Ofertas

- Quienes Somos
- Perfil
- Cerrar Sesión

Usuario Administrador

- Perfil
- Ofertas
- Archivos
- Administrador
- Cerrar Sesión

21/05/2021

- Se crea el componente AdminBar, el cual es cargado unicamente por los componentes que lo necesiten, así que no tiene variables de entrada
- Se modifica el AdminBar para que funcione con NextJs
- Se agregan los menús Sistema, Usuarios y ofertas
- La página de inicio "/" en caso de iniciar sesión como administrador, muestra el dashboard en el menú de configurar Sistema

21/05/2021

- Se agregan componentes a la página de inicio para cuando el usuario es administrador
- Bajo el menú Sistema/Configuración se despliega el formulario que permite editar las configuraciones relacionadas con el SEO
- se agregan los campos relacionados con el backend:
 - **Nombre de la empresa:** Utilizado en el sistema SEO
 - **Nombres del propietario:** Utilizado en el formulario de contacto
 - **Apellidos del propietario:** Utilizado en el formulario de contacto
 - **Descripción del Negocio:** Utilizado en el sistema SEO y formulario de contacto
 - **Numero de contacto 1:** Utilizado en el sistema SEO y formulario de contacto
 - **Numero de contacto 2:** (Opcional) Utilizado en el sistema SEO y formulario de contacto
 - **Dirección del negocio:** Utilizado en el sistema SEO, Formulario de contacto y localización
 - **Correo Electronico 1:** Utilizado en el sistema SEO, Formulario de contacto y localización
 - **Correo Electronico 2:** (opcional) Utilizado en el sistema SEO, Formulario de contacto y localización
- Agregado submenú de seguridad al menú de Sistema
- Agregada tabla de configuración general del Sistema

Archivo	Tipo	Función
models/sistema.model.js	modelo	Tabla de configuración del sistema
routes/admin.routes.js	rutas	Ruta de llamadas de api para usuarios con rol de administración
controllers/admin.controller.js	controlador	Controlador para las llamadas de api mediante la ruta admin.routes.js

22/05/2021

- Reconstruido sistema de comunicacion con la API

Headers

nombre	valor
Content-Type	application/json
x-access-token	token

Body

tipo: raw

valor

```
{
  "donde": "", //Lista de columnas en caso de busqueda, separadas por
signo igual y caracter de tuberia (ex: id_usuario=1|nombre="juanito
piguave")
  "limite": "", //funciona normal
  "off": "", //cuantos cosas por pagina
  "orden": "id_sistema DESC" //ordena descendente o ascendente por nombre
de columna, no olvidar llenar este campo para no generar error
}
```

26/05/2021

- reconstruccion del sistema de API fase 2

Nombre del Campo	Valor	Descripcion
columnas	["col1","col2","col3"]	Lista de columnas que se desea obtener en la consulta
donde	[["col","=","val"],["col2","=","val2"],["col3","=","val"]]	Array de arrays conteniendo el nombre de la columna, el operador y el valor
orden	["col","asc/desc"]	Array con el nombre de la columna y el operador de orden (asc/desc)
limite	["page","offset"]	Array con el numero de pagina y el numero de offset

27/05/2021

- Reconstruccion del sistema de API fase 3
- Los llamados a la api dependen del backend, de esta manera, el CRUD queda asi:

Tipo de llamado	Rol del usuario	Descripcion
Ver_[x]	Admin	Este llamado permite obtener una lista de elementos de la base de datos, se usa para la edicion de datos y no requiere seleccion de las columnas

28/05/2021

- Reconstruccion del sistema de API fase 4
- se separa la interfaz grafica de las paginas en NextJS en componentes que se ocupan de toda su logica
- los componentes heredan los datos segun sean llamados por los roles de admin o user
- las paginas deciden que componente cargar segun el estado de la sesion y el tipo de usuario que hace el llamado
- las operaciones CRUD dependen del backend

29/05/2021

- Reconstruccion del sistema de API fase 5
- completado sistema de lectura de datos del backend
- completado sistema de escritura/sobreescritura de datos del backend
- completado sistema modular de componentes
- completado sistema modular de rutas y paginas para nextjs
- completado sistema de gestion de estados para los elementos de formularios de los componentes
- desarrollo temporal de la pagina de inicio para usuarios sin sesion.

30/05/2021

- Actualizado el NavBar
 - al iniciar la carga de una pagina, se consulta la api publica info_sistema, para obtener el nombre de la empresa, y otros datos
- Creada la pagina de Informacion sobre el negocio
- Creada la pagina de contacto
- Creado el componente de Informacion sobre el negocio
- Creado el componente de formulario de contacto
 - Agregado el modelo de contactos
 - Agregado el formato para agregar datos
 - Se registra la direccion ip de la persona interesada

Finalizada la fase 1 (vease [Fases del proyecto](#))

31/05/2021

- Agregada la pantalla ver_formularios
- Agregado el componente ver_formularios
- Modificado el navbar, el menu sistema, ahora solo contiene los siguientes elementos
 - Configuracion
 - Seguridad

- Solicitudes de informacion
- Creada la lista de elementos que constituyen la pantalla de solicitudes de informacion

1/06/2021

- Agregado componente de paginacion

2/06/2021

- Eliminado componente de paginacion
- Agregado control de busqueda, seleccion de columnas, orden y limite en el backend
- Error al intentar cargar inicialmente un estado

3/06/2021

- Corregido problema que impedia la carga de estado inicial en el componente ver_formularios
- Avanzado el protocolo de consulta de api, se mantiene un formato basico de consultas, solo con igualdades
- se ha encontrado un error que obliga a presionar el boton de busqueda dos veces puesto que la variable "Donde" no se carga con los datos actualizados del formulario

4/06/2021

- Correcciones menores a la pantalla de ver solicitudes de contacto
- Agregado sistema de busqueda
- no se agregara sistema de filtro ni orden

5/06/2021

- Completado el sistema de paginacion
- completado el sistema de busqueda basica
- completado el sistema de lectura modal
- **PENDIENTE PARA LA FASE 4:** agregar un sistema de respuesta por correo electronico
- **PENDIENTE PARA LA FASE 4:** agregar un sistema de estado para indicar si un mensaje ha sido leído o no

6/06/2021

- completado el sistema de borrado de la pagina ver_informacion_contacto
- Creada la pagina configuracion de seguridad dentro del directorio Seguridad
- Creada la llamada de API para cambiar la contraseña del usuario administrador
- Creada la llamada de API para cambiar la contraseña del usuario
- Creada la pagina configuracion de seguridad dentro del directorio Seguridad para el usuario normal
- Creada la pagina de lista de cuentas de usuario
 - Pendiente agregar funciones basicas de CRUD

7/06/2021

- Creado subcomponente de edicion modal para la administracion de cuentas

- Creada la ruta de crear_cuenta
- Creado el componente de crear_cuenta

8/06/2021

- Agregado componente nodemailer
- Agregado el modelo sendmail
- Agregado el controlador ver_confsendmail
- Agregado el controlador actualizar_confsendmail
- Agregada opcion de menu para las configuraciones de envios de correos en el adminbar
- **PENDIENTE PARA LA FASE 4** agregar un sistema WYSIWYG para la configuracion de envios de correos
- Completado el sistema de configuracion de envio de correos
- **PENDIENTE PARA LA FASE 4** agregar mas campos de mensajes para la configuracion de envios de correos
- **PENDIENTE IMPORTANTE** agregar datos iniciales para configuraciones de sistemas y configuraciones de envio de correos
- Completado agregar nuevas cuentas de usuario
 - pendiente actualizar cuentas de usuario, esto incluye la opcion de desactivarlas por estado

9/06/2021

- Completada la pantalla de gestion de usuarios
- Eliminada entrada del menu de administrador "Galeria de Imagenes"
- Creada la pagina "archivos"
 - el sistema de gestion de archivos permite enviar y recibir archivos por usuario de la siguiente manera:
- Eliminada la pagina de archivos a cambio de mensajes del sistema
- Eliminada entrada del menu archivos
- Creada entrada del menu, mensajes salientes
- Creada entrada del menu, mensajes entrantes **OBSERVACIONES** aun no estoy seguro de si deba crear una galeria de archivos o en su lugar, un sistema de mensajeria basada en el correo electronico, pero probablemente esta sea la mejor opcion. En caso de optar por esta ultima, es necesario crear una bandeja saliente y otra entrante y estaria terminada la fase 2 del proyecto

La fase 3 consiste en agregar las interacciones del usuario con el dashboard, para esto, se creara un nuevo usuario con su respectiva contraseña. las paginas propuestas para el dashboard son las siguientes:

- inicio: gestion detalles del usuario como datos personales, tambien es posible que ponga en su lugar la pagina de mensajes entrantes a manera de notificaciones
- transacciones?(fase 5): el usuario podra ver todas las cosas que ha comprado y ver si su solicitud ha sido atendida
- sistema memorial?(fase 4): el usuario podra ver el perfil de su familiar fallecido, agregar eventos o comentarios a manera de subpost. cada una de estas cosas debera ser moderada por el administrador

La fase 4 consiste en crear el sistema de memorial en si. Este sistema crea perfiles para las personas fallecidas y deberan ser creados por los administradores, asociados a un usuario. muchos perfiles -> un

usuario

La fase 5 consiste en hacer pruebas de funcionamiento y agregar alguna funcion que se haya sido recomendada

La fase 6 consiste en hacer pruebas de uso

10/06/2021

- Cambiada la opcion de menu de Mensajes salientes a Mensajes
- Eliminada la opcion de menu de mensajes entrantes
- Creada la pagina de mensajes
- crear tabla de mensajes
- crear tabla de archivos
- Crear el modelo de mensajes
 - mensajes relacionado con usuarios
- Crear el modelo de archivos
- Crear el modelo de galerias
- Agregada la pagina archivos
- Agregada la pagina galeria
- Agregado componente enviar_mensaje
- Agregado la ruta enviar_mensaje
- Agregado el controlador enviar_mensaje

Mensajes

Se almacenan los mensajes que son mostrados al usuario a manera de notificaciones. El usuario los podra consultar en cualquier momento

nombre del campo	tipo	comentarios
id_mensaje	integer	
asunto	string	
contenido	text	
adjuntos	text	en formato json, lista de archivos
id_usuario	integer	a quien va dirigido el mensaje
etiquetas	text	
estado	string	

Archivos Esta tabla almacena las rutas a los archivos privados que se guardaran exclusivamente para sus receptores

nombre del campo	tipo	comentarios
id_archivo	integer	
nombre	string	nombre actual del archivo, sin extension

nombre del campo	tipo	comentarios
nombre_original	string	nombre original del archivo, con extension
descripcion	text	
ext	string	
size	string	en kb
estado	string	
id_usuario	integer	quien puede ver el archivo aparte del administrador

Galeria Esta tabla almacena los archivos en la ruta publica del sistema, que sirvan para ser visualizados sin necesidad de algun rol en especifico.

nombre del campo	tipo	comentarios
id_archivo	integer	
nombre	string	nombre actual del archivo, sin extension
nombre_original	string	nombre original del archivo, con extension
descripcion	text	
ext	string	
size	string	en kb
estado	string	

11/06/2021

- Creada la funcion de archivos para los mensajes
- **PENDIENTE PARA LA FASE 4:** Crear un gestor de archivos
- **PENDIENTE PARA LA FASE 4:** Crear un gestor de galerias
- **PENDIENTE PARA LA FASE 4:** Permitir adjuntar multiples archivos

Finalizada la fase 2 (vease [Fases del proyecto](#))

12/06/2021

- Agregada la funcion de enviar correo electronico paralelo a las notificaciones del sistema
- Creado controlador de guardar_archivos
- Completada la funcion de envio de mensajes
- Planificando el componente ver_notificaciones
- Planificando el formato de visualizacion de usuario normal
- Actualizados enlaces del navbar para usuario normal
- Enlazado el componente quienesomos a la pagina "quienes somos" para el usuario normal
- Enlazado el componente quienesomos a la pagina "quienes somos" para el usuario administrador
- Enlazado el componente contacto a la pagina "contactenos" para el usuario normal
- Enlazado el componente contacto a la pagina "contactenos" para el usuario administrador

12/06/2021

- Creado el componente inicio para el usuario normal
- Completada la pagina de inicio/ver mensajes para el usuario normal
- pendiente: visualizar archivos por usuario
- agregado sistema de descarga de archivos mediante autenticacion jwt

13/06/2021

- Completado sistema de descarga de archivos mediante autenticacion jwt
- Creado el componente para la pagina /Sistema/seguridad

14/06/2021

- Creado el componente userbar
- Creada la ruta de actualizar_password
- Creado el controlador de actualizar_password
- Completado el sistema de actualizacion de contraseña

15/06/2021

- Creada pagina de perfil para el usuario normal
- Creado controlador para actualizar perfil de usuario normal
- Creada ruta para actualizar perfil de usuario normal

Finalizada la fase 3 (vease [Fases del proyecto](#))

- Removida la opcion del menu ofertas para el usuario normal

16/06/2021

- Actualizacion de las fases del proyecto
- Actualizacion del Navbar
- Actualizacion del Userbar
- Actualizacion del Adminbar

17/06/2021

- Modelado del sistema Memorial
- Creado el modelo "Memorias"
- Creado el modelo Fotos
- Creado el modelo Galerias
- Creada la relacion fotos_memorias
- Creada la relacion memorias, fotos, galerias, usuarios
- Creada pagina memorial en el directorio /Memorial/ver_memorias
- Creado el componente ver_memorias
- Creada la ruta de administrador /ver_memorias
- Creado el controlador de administrador /ver_memorias

Memorias Tabla para registrar las entradas del sistema memorial. Esta tabla permite al administrador crear una plantilla con los datos que posea, pero el usuario es quien podrá actualizarla segun crea conveniente

nombre del campo	tipo	comentarios
id_memoria	integer	
titulo	string	eliminado
url	string	
contenido	text	
id_foto	integer	relacionado con la tabla de fotos
id_galeria	relacionado con la tabla de galerias	
id_usuario	relacionado con la tabla de usuarios	
fecha_muerte	datetime	
causa_muerte	text	

Galerias Tabla para agrupar las fotos en galerias

nombre del campo	tipo	comentarios
id_galeria	integer	
nombre	string	
descripcion		

fotos_galerias tabla para relacion n:n entre fotos y galerias

nombre del campo	tipo	comentarios
id_galeria	integer	
id_foto	integer	

fotos Esta tabla difiere de la de archivos, porque de esta se almacena solo rutas publicas

nombre del campo	tipo	comentarios
id_foto	integer	
nombre	string	nombre actual del archivo, sin extension
nombre_original	string	nombre original del archivo, con extension
descripcion	text	
ext	string	
size	string	en kb
estado	string	

18/06/2021

- Creada la pagina [slug].js para busqueda parametrizada de memorias

19/06/2021

- Creado el componente publico para la visualizacion del perfil memorial
- Modificado el componente de cuentas.js
 - agregada opcion del menu Agregar Memoria
- Corregido modelo de memorias
 - agregado el campo fecha_nacimiento
 - eliminado el campo razon_muerte
- Creada la ruta de administrador crear_memoria
- Creado el controlador de administrador crear_memoria
- Creada la ruta de administrador "guardar_foto"
- Creado el componente de administrador "guardar_foto"
- Corregido modelo fotos
- Corregido la relacion entre memorias y usuarios y fotos

20/06/2021

Sin avances por circunstancias de fuerza mayor

21/06/2021

- Creada la plantilla de ver_memoria
- Creada la ruta publica para ver_foto
- Creado controlador publico para ver_foto
- Creado el componente publico de ver_memorias

22/06/2021

- Inclusion del componente ver_memorias en la pagina publica ver_memorias
- Completada la pagina de ver_memorias
- modificado el navbar para mostrar la pagina de ver_memorias

23/06/2021

- Actualizado controlador de administrador para ver_memorias
- Actualizada pagina de administrador para ver_memorias
- Eliminado nodo html ModalData.archivo.nombre_original de la pagina inicio para usuario normal
- Eliminado nodo html ModalData.createdAt de la pagina inicio para usuario normal

Finalizada la fase 4 (vease [Fases del proyecto](#))

Pendiente para mañana

- crear el panel de administracion de usuario para las memorias
- reparar pantalla de contactos y quienes somos para usuarios publicos

Pendiente:

- crear un mensaje inicial de bienvenida a los usuarios del sistema
- crear una carga inicial para las configuraciones del sistema y para el usuario administrador(entradas iniciales en las respectivas tablas)
- crear una explicacion del funcionamiento del sistema de plantillas y componentes para las pantallas publicas
- crear una pagina para las configuraciones de las entradas del sistema memorial
- crear un componente para crear,visualizar,editar y eliminar entradas del sistema memorial
 - terminar pantalla de ver_memorias
 - crear componente actualizar memorias
 - crear componente generar url
 - crear componente generar codigo qr
 - crear componente ver memorias para el usuario
 - crear componente actualizar memorias para el usuario
 - crear componente previsualizar memorias para el usuario
 - crear pagina parametrizada de ver memorias
- Crear carga inicial para una foto predeterminada
- Actualizar modulo del sistema, para agregar funcion de base_url del sistema y base_url de la api

Propuestas de mejora

- Crear un sistema de notificaciones mediante websockets
- Crear un sistema de mensajeria instantanea
- Crear un sistema de respuesta automatica
- https://owasp.org/www-project-top-ten/?fbclid=IwAR0i5dKgKGrah8RvuVnJ1bvf6ckJGoM_u31LYJKqiFcXgQ2oVyDbOit3q2Y
- Crear un manual para la creacion de nuevos componentes para el sistema
- Crear un manual para la creacion de un sistema de compras en linea
- Crear un sistema automatizado para agregar sistemas de metricas basados en google o facebook
- Crear nuevos campos para personalizar los mensajes segun las acciones del sistema
- Crear un componente para gestionar la lista de mensajes enviados
- Crear un componente para gestionar la lista de memorias creadas por el administrador
- Permitir al usuario crear galerias de fotos de sus personas fallecidas
- Permitir al usuario crear eventos
- Actualizar el layout de forma que permita a las redes sociales cargar una vista previa coherente