

MANUAL DEL ADMINISTRADOR DEL SISTEMA / DEVO'S

Para poder trabajar con el sistema bypsa el servidor que se monte ya sea Linux o Windows Server deberá tener.

- Angular CLI,
- Node
- NPM
- Mongo DB

Una vez instalada las librerías necesarias para poder correr los comandos en cada una de las instancias.

CONFIGURACIONES DEL SISTEMA

Para poder configurar el sistema se debe hacer de dos formas:

El sistema se divide en 3 partes, (FrontEnd – Angular, Backend – Node, Express y con una base de datos en Mongo DB).

Una vez descargado cada repositorio se deberá lanzar el comando **npm i** para poder instalar las dependencias con las que se desarrollo el sistema.

BACKEND

Los archivos nodemailer.json.default y app.json.default son solo archivos de muestra para lo cual se debe sacar una copia y renombrarlos por nodemailer.json y app.json respectivamente.

EL archivo nodemailer.json, es la configuración de un correo (Gmail, Hotmail o de empresa para usar los servicios de envío , si utiliza el correo de empresa recordar el manejo de servidor de entrada salida buscar documentación para ello en Node o Express como librerías externas).

El archivo app.json es la configuración de dirección de base de datos mongodb

Crear process.env.PORT puerto de entorno de producción para el escucha del servicio por defecto esta en el puerto 3001

Para poder acceder a la api de la aplicación es el "HOST:3001"
AMBIENTE local <http://127.0.0.1:3001>

AMBIENTE externo <http://IP:3001> o <http://midominio.com:3001>

INICIAR EL PROYECTO

Los siguientes comandos son para que el proyecto inicie en el servidor.

- npm run deploy: Este comando levanta todos los valores iniciales que necesita la aplicación tales como Ciudad, Rol, Usuario entre otros (una vez finalizado el comando presionar ctrl + c para terminar la ejecución).
- Npm run start: este comando es para iniciar la aplicación con ello el sistema backend ya esta operativo.

MODULOS

Los módulos del sistema están basados en el patrón de diseño MVC por lo para acceder a un modulo es <http://midominio.com:3001/CONTROLADOR/ACCION>

/CONTROLADOR/ACCION = Es la ruta del endpoint dentro del entorno de nodejs

CONTROLOADOR = nombre del modulo en singular(ejemplo "user").

ACCION = nombre de la acción del modulo (ejemplo "list").

Y METODO el cual puede ser (get, post, delete) y depende de la ruta el endpoint.

Ejemplo completo (<http://127.0.0.1:3001/user/list>, metodo post)

DESCRIPCION DE LOS MODULOS

MODULO ACCOUNT

1. **/account/list, método get :** Es el endpoint que muestra el listado de los registros
2. **/account/view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/account/add, método post :** Es el endpoint que agrega un registro
4. **/account/edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/account/delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id
6. **/account /addaccountimg, método post:** Es el endpoint para subir una imagen para los registros
7. **/ account /deleteaccountimg/:name, método delete:** Es el endpoint para eliminar una imagen para los registros basado en el nombre

MODULO ALTERNATIVE

1. **/ alternative /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ alternative /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ alternative /add, método post :** Es el endpoint que agrega un registro
4. **/ alternative /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ alternative /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO annexMedicalBusiness

1. **/ annexMedicalBusiness /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ annexMedicalBusiness /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id

3. **/ annexMedicalBusiness /add, método post :** Es el endpoint que agrega un registro
4. **/ annexMedicalBusiness /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ annexMedicalBusiness /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO annexMedicalBusinessItem

1. **/ annexMedicalBusinessItem /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ annexMedicalBusinessItem /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ annexMedicalBusinessItem /add, método post :** Es el endpoint que agrega un registro
4. **/ annexMedicalBusinessItem /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ annexMedicalBusinessItem /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO apiClient

1. **/ apiClient / login, método get :** Es el endpoint de inicio de sesión de clientes
2. **/ apiClient / recoveryPassword /:id, método get :** Es el endpoint de recuperación de contraseña en específico basado en el id
3. **/ apiClientPolicy / list, método post :** Es en endpoint que muestra el listado de pólizas de los registros de un cliente
4. **/ apiClientWallet / list /:id, método post:** Es el endpoint que muestra el listado de carteras de un cliente

MODULO authorizationTime

1. **/ authorizationTime /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ authorizationTime /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ authorizationTime /add, método post :** Es el endpoint que agrega un registro
4. **/ authorizationTime /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id

5. **/ authorizationTime /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO BANCO

1. **/ bank /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ bank /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ bank /add, método post :** Es el endpoint que agrega un registro
4. **/ bank /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ bank /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO bankInsurance

1. **/ bankInsurance /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ bankInsurance /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ bankInsurance /add, método post :** Es el endpoint que agrega un registro
4. **/ bankInsurance /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ bankInsurance /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO billing

1. **/ billing /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ billing /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ billing /add, método post :** Es el endpoint que agrega un registro
4. **/ billing /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ billing /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO billingPolicy

1. **/ billingPolicy /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ billingPolicy /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ billingPolicy /add, método post :** Es el endpoint que agrega un registro
4. **/ billingPolicy /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ billingPolicy /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO binnacleExpirationDate

1. **/ binnacleExpirationDate /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ binnacleExpirationDate /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ binnacleExpirationDate /add, método post :** Es el endpoint que agrega un registro
4. **/ binnacleExpirationDate /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ binnacleExpirationDate /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO branch

1. **/ branch /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ branch /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ branch /add, método post :** Es el endpoint que agrega un registro
4. **/ branch /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ branch /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO business

1. **/ business /list, método get :** Es el endpoint que muestra el listado de los registros

2. **/ business /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ business /add, método post :** Es el endpoint que agrega un registro
4. **/ business /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ business /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id.

MODULO businessClient

1. **/ businessClient /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ businessClient /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ businessClient /add, método post :** Es el endpoint que agrega un registro
4. **/ businessClient /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ businessClient /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO car

1. **/ car /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ car /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ car /add, método post :** Es el endpoint que agrega un registro
4. **/ car /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ car /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO carBrand

1. **/ carBrand /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ carBrand /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ carBrand /add, método post :** Es el endpoint que agrega un registro

4. **/ carBrand /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ carBrand /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO carColor

1. **/ carColor /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ carColor /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ carColor /add, método post :** Es el endpoint que agrega un registro
4. **/ carColor /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ carColor /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO carModel

1. **/ carModel /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ carModel /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ carModel /add, método post :** Es el endpoint que agrega un registro
4. **/ carModel /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ carModel /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO carType

1. **/ carType /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ carType /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ carType /add, método post :** Es el endpoint que agrega un registro
4. **/ carType /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ carType /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO city

1. **/ city /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ city /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ city /add, método post :** Es el endpoint que agrega un registro
4. **/ city /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ city /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO clearanceTime

1. **/ clearanceTime /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ clearanceTime /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ clearanceTime /add, método post :** Es el endpoint que agrega un registro
4. **/ clearanceTime /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ clearanceTime /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO CLIENTE

1. **/client/list, método get :** Es el endpoint que muestra el listado de los registros
 2. **/client/view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
 3. **/client/add, método post :** Es el endpoint que agrega un registro
 4. **/client/edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
 5. **/client/delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id
 6. **/client/addclientImg, método post:** Es el endpoint para subir una imagen para los registros
- /client/deleteclientImg/:name, método delete:** Es el endpoint para eliminar una imagen para los registros basado en el nombre

MODULO country

1. **/ country /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ country /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ country /add, método post :** Es el endpoint que agrega un registro
4. **/ country /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ country /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO creditNote

1. **/ creditNote /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ creditNote /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ creditNote /add, método post :** Es el endpoint que agrega un registro
4. **/ creditNote /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ creditNote /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO deductible

1. **/ deductible /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ deductible /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ deductible /add, método post :** Es el endpoint que agrega un registro
4. **/ deductible /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ deductible /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO dependent

1. **/ dependent /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ dependent /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ dependent /add, método post :** Es el endpoint que agrega un registro
4. **/ dependent /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ dependent /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id
6. **/ dependent / adddependentImg, método post:** Es el endpoint para subir una imagen para los registros
7. **/ dependent / deletedependentImg /:name, método delete:** Es el endpoint para eliminar una imagen para los registros basado en el nombre

MODULO deprecation

1. **/ deprecation /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ deprecation /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ deprecation /add, método post :** Es el endpoint que agrega un registro
4. **/ deprecation /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ deprecation /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO frequencyPayment

1. **/ frequencyPayment /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ frequencyPayment /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ frequencyPayment /add, método post :** Es el endpoint que agrega un registro
4. **/ frequencyPayment /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id

5. **/ frequencyPayment /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO helpLink

1. **/ helpLink /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ helpLink /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ helpLink /add, método post :** Es el endpoint que agrega un registro
4. **/ helpLink /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ helpLink /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO income

1. **/ income /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ income /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ income /add, método post :** Es el endpoint que agrega un registro
4. **/ income /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ income /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO insurance

1. **/ insurance /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ insurance /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ insurance /add, método post :** Es el endpoint que agrega un registro
4. **/ insurance /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ insurance /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id
6. **/ insurance / addinsuranceimg, método post:** Es el endpoint para subir una imagen para los registros

7. **/ insurance / deleteinsurancetimg /:name, método delete:** Es el endpoint para eliminar una imagen para los registros basado en el nombre

MODULO issue

1. **/ issue /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ issue /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ issue /add, método post :** Es el endpoint que agrega un registro
4. **/ issue /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ issue /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO itemAnnex

1. **/ itemAnnex /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ itemAnnex /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ itemAnnex /add, método post :** Es el endpoint que agrega un registro
4. **/ itemAnnex /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ itemAnnex /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO itemAnnexExtra

1. **/ itemAnnexExtra /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ itemAnnexExtra /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ itemAnnexExtra /add, método post :** Es el endpoint que agrega un registro
4. **/ itemAnnexExtra /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ itemAnnexExtra /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO letterAccident

1. **/ letterAccident /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ letterAccident /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ letterAccident /add, método post :** Es el endpoint que agrega un registro
4. **/ letterAccident /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ letterAccident /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO log

1. **/ log/list, método get :** Es el endpoint que muestra el listado de los sucesos del sistema.

MODULO maritalStatus

1. **/ maritalStatus /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ maritalStatus /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ maritalStatus /add, método post :** Es el endpoint que agrega un registro
4. **/ maritalStatus /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ maritalStatus /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO noRenewal

1. **/ noRenewal /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ noRenewal /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ noRenewal /add, método post :** Es el endpoint que agrega un registro
4. **/ noRenewal /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id

5. **/ noRenewal /delete/:id, método delete** : Es el endpoint que elimina un registro basado en el id

MODULO noRenewalRamo

1. **/ noRenewalRamo /list, método get** : Es el endpoint que muestra el listado de los registros
2. **/ noRenewalRamo /view/:id, método get** : Es el endpoint que visualiza un registro en específico basado en el id
3. **/ noRenewalRamo /add, método post** : Es el endpoint que agrega un registro
4. **/ noRenewalRamo /edit/:id, método post**: Es el endpoint que edita un registro en específico basado en el id
5. **/ noRenewalRamo /delete/:id, método delete** : Es el endpoint que elimina un registro basado en el id

MODULO param

1. **/ param/list, método get** : Es el endpoint que muestra el listado de parámetros definidos para el sistema en el archivo helper.json en base a clave valor

MODULO paymentType

1. **/ paymentType /list, método get** : Es el endpoint que muestra el listado de los registros
2. **/ paymentType /view/:id, método get** : Es el endpoint que visualiza un registro en específico basado en el id
3. **/ paymentType /add, método post** : Es el endpoint que agrega un registro
4. **/ paymentType /edit/:id, método post**: Es el endpoint que edita un registro en específico basado en el id
5. **/ paymentType /delete/:id, método delete** : Es el endpoint que elimina un registro basado en el id

MODULO percentageRamo

1. **/ percentageRamo /list, método get** : Es el endpoint que muestra el listado de los registros
2. **/ percentageRamo /view/:id, método get** : Es el endpoint que visualiza un registro en específico basado en el id
3. **/ percentageRamo /add, método post** : Es el endpoint que agrega un registro

4. **/percentageRamo /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/percentageRamo /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO pickup

1. **/pickup /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/pickup /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/pickup /add, método post :** Es el endpoint que agrega un registro
4. **/pickup /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/pickup /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO plan

1. **/plan /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/plan /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/plan /add, método post :** Es el endpoint que agrega un registro
4. **/plan /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/plan /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO planAlternative

1. **/planAlternative /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/planAlternative /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/planAlternative /add, método post :** Es el endpoint que agrega un registro
4. **/planAlternative /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/planAlternative /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO planAssociation

1. **/ planAssociation /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ planAssociation /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ planAssociation /add, método post :** Es el endpoint que agrega un registro
4. **/ planAssociation /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ planAssociation /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO policy

1. **/ policy /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ policy /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ policy /add, método post :** Es el endpoint que agrega un registro
4. **/ policy /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ policy /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO policyAnnex

1. **/ policyAnnex /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ policyAnnex /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ policyAnnex /add, método post :** Es el endpoint que agrega un registro
4. **/ policyAnnex /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ policyAnnex /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id
6. **/ policyAnnex / addpolicyAnnexImg, método post:** Es el endpoint para subir una imagen para los registros

7. **/ policyAnnex / deletepolicyAnnexImg /:name, método delete:** Es el endpoint para eliminar una imagen para los registros basado en el nombre

MODULO policyType

1. **/ policyType /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ policyType /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ policyType /add, método post :** Es el endpoint que agrega un registro
4. **/ policyType /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ policyType /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO quote

1. **/ quote /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ quote /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ quote /add, método post :** Es el endpoint que agrega un registro
4. **/ quote /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ quote /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO ramo

1. **/ ramo /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ ramo /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ ramo /add, método post :** Es el endpoint que agrega un registro
4. **/ ramo /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ ramo /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO renewal

1. **/ renewal /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ renewal /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ renewal /add, método post :** Es el endpoint que agrega un registro
4. **/ renewal /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ renewal /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO role

1. **/ role /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ role /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ role /add, método post :** Es el endpoint que agrega un registro
4. **/ role /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ role /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO route

1. **/ route /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ route /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ route /add, método post :** Es el endpoint que agrega un registro
4. **/ route /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ route /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO setting

1. **/ setting /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ setting /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id

3. **/ setting /add, método post :** Es el endpoint que agrega un registro
4. **/ setting /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ setting /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO sinister

1. **/ sinister /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ sinister /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ sinister /add, método post :** Es el endpoint que agrega un registro
4. **/ sinister /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ sinister /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO sinisterCar

1. **/ sinisterCar /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ sinisterCar /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ sinisterCar /add, método post :** Es el endpoint que agrega un registro
4. **/ sinisterCar /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ sinisterCar /delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO sinisterDocumentation

1. **/ sinisterDocumentation /list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ sinisterDocumentation /view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ sinisterDocumentation /add, método post :** Es el endpoint que agrega un registro
4. **/ sinisterDocumentation /edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id

5. **/ sinisterDocumentation /delete/:id, método delete** : Es el endpoint que elimina un registro basado en el id

MODULO sinisterGeneral

1. **/ sinisterGeneral /list, método get** : Es el endpoint que muestra el listado de los registros
2. **/ sinisterGeneral /view/:id, método get** : Es el endpoint que visualiza un registro en específico basado en el id
3. **/ sinisterGeneral /add, método post** : Es el endpoint que agrega un registro
4. **/ sinisterGeneral /edit/:id, método post**: Es el endpoint que edita un registro en específico basado en el id
5. **/ sinisterGeneral /delete/:id, método delete** : Es el endpoint que elimina un registro basado en el id

MODULO sinisterGeneralDocumentation

1. **/ sinisterGeneralDocumentation/list, método get** : Es el endpoint que muestra el listado de los registros
2. **/ sinisterGeneralDocumentation/view/:id, método get** : Es el endpoint que visualiza un registro en específico basado en el id
3. **/ sinisterGeneralDocumentation/add, método post** : Es el endpoint que agrega un registro
4. **/ sinisterGeneralDocumentation/edit/:id, método post**: Es el endpoint que edita un registro en específico basado en el id
5. **/ sinisterGeneralDocumentation/delete/:id, método delete** : Es el endpoint que elimina un registro basado en el id

MODULO tasa

1. **/ tasa/list, método get** : Es el endpoint que muestra el listado de los registros
2. **/ tasa/view/:id, método get** : Es el endpoint que visualiza un registro en específico basado en el id
3. **/ tasa/add, método post** : Es el endpoint que agrega un registro
4. **/ tasa/edit/:id, método post**: Es el endpoint que edita un registro en específico basado en el id
5. **/ tasa/delete/:id, método delete** : Es el endpoint que elimina un registro basado en el id

MODULO typeClient

1. **/ typeClient/list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ typeClient/view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ typeClient/add, método post :** Es el endpoint que agrega un registro
4. **/ typeClient/edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ typeClient/delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO user

1. **/ user/list, método get :** Es el endpoint que muestra el listado de los registros
6. **/ user/view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
7. **/ user/add, método post :** Es el endpoint que agrega un registro
8. **/ user/edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
9. **/ user/delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO wallet

1. **/ wallet/list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ wallet/view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ wallet/add, método post :** Es el endpoint que agrega un registro
4. **/ wallet/edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ wallet/delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO walletPayment

1. **/ walletPayment/list, método get :** Es el endpoint que muestra el listado de los registros

2. **/ walletPayment/view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ walletPayment/add, método post :** Es el endpoint que agrega un registro
4. **/ walletPayment/edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ walletPayment/delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

MODULO walletPaymentBinnacle

1. **/ walletPaymentBinnacle/list, método get :** Es el endpoint que muestra el listado de los registros
2. **/ walletPaymentBinnacle/view/:id, método get :** Es el endpoint que visualiza un registro en específico basado en el id
3. **/ walletPaymentBinnacle/add, método post :** Es el endpoint que agrega un registro
4. **/ walletPaymentBinnacle/edit/:id, método post:** Es el endpoint que edita un registro en específico basado en el id
5. **/ walletPaymentBinnacle/delete/:id, método delete :** Es el endpoint que elimina un registro basado en el id

FRONTEND

INICIAR EL PROYECTO

Los siguientes comandos son para que el proyecto inicie en el servidor.

- Ng serve o npm run start: Este comando es para iniciar la aplicación con ello el sistema Front ya este operativo en el puerto 4200, este puerto es por default para la fase dev. En este apartado podrán realizar pruebas y cambios en tiempo real para poder tener una mejor percepción del reload en tiempo real de los cambios que se realicen.
- Ng build o npm run build: Este comando ya comprime todo lo necesario para entregarnos un archivo final de html, css y Js los cuales ya podremos subir a nuestro server, recordar que se genera una carpeta llamada dist donde nos depositara los

archivos antes mencionados.(comentar todos los console.log de los ts).

Al subir los archivos ya cambia el puerto al 80 con e que se trabaja por defecto.

DISTRIBUCIÓN DEL SISTEMA FRONTEND (ANGULAR)

Todo lo que se necesita saber de la app y sus librerías esta en el package.json

Todos los archivos que deseen cambiar o crear nuevos módulos lo deben hacer en el src/app

(La forma de registrar y demás opción para un nuevo componente se debe hacer basado en manifest de Angular)

<https://angular.io>

La distribución es:

- Components: son donde se podrán los componentes de cada componente padre.
- Theme: es la base del proyecto que se divide en:
 - Components: son los componentes del template para crear la estructura del sistema
 - Directives y Pipes (ver documentación Angular)
 - Styles: el archivo esta creado en Sass y en cada archivo esta lo necesario para cada componente con as variables del proyecto.
- Providers: Es para los servicios externos con el backend.
- Pages: es por cada vista una page donde pueden existir cada componente de vista.
- Assest: Lugar donde se pondrá js, css e imágenes que necesite el sistema.
- Config: en su interior esta Project-config.ts, aquí esta la url del servidor en node (backend) y los mensajes del sistema para las acciones que se realizan en el mismo.

- Index.html es el archivo único de carga del sistema, aquí es donde se carga el efecto de loading.

Nota: es importante recordar que existe los scss globales y por componentes.

Leer la documentación de Angular para registrar su nuevo componente si el IDE no lo hace por usted.

Esta basado en Angular 4 todo el desarrollo en dicho momento. Para la documentación entre Angular 4 o 6 los cambios no son tan radicales.

CEO Wilson Pérez
Cintanegra