DOCUMENTACIÓN TÉCNICA OT N°1

**Detalles de la Documentación**

|  |  |
| --- | --- |
| **Tipo de Documentación:** | Documentación Técnica |
| **Cliente:** | MITIC |
| **Interesados:** | OEE |
| **Autores:** | Freelancers del Paraguay |
| **Fecha de Entrega:** | 19/01/2023 |

**Versiones y Modificaciones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Autor** | **Modificaciones** |
| 1.0 | 06/12/2023 | Freelancers del Paraguay | Inicio |
| 2.0 | 26/12/2023 | Freelancers del Paraguay | Base de datos, recursos necesarios |
| 3.0 | 19/01/2023 | Freelancers del Paraguay | Base de datos. |

[**Detalles del Sistema 2**](#_wrrt1vxerqpk)

[**Funcionalidades globales 2**](#_4d34og8)

[**Recursos necesarios 3**](#_s0z9ruu0o78s)

[**Despliegue backend 3**](#_lu6tjvqk0nnv)

[**Despliegue Frontend 11**](#_jmzuu944wnz8)

[**Diccionario de Datos 14**](#_a10scov0su6f)

[**Modificaciones de Base de Datos 16**](#_a7xmli38jfyc)

# Detalles del Sistema

Stack tecnológico

* **Backend**
  + JAVA 11
  + MAVEN 3.8.4
  + SPRING BOOT 2.7.5
  + REDIS 5.0
* **Frontend**
  + ANGULAR 13.1.1
  + ULTIMA PRIMENG FROM PRIMEFACES 13.1.1
  + NODE 16.17
  + NPM 8.15
* **Base de datos**
  + POSTGRESQL 14.10

# Funcionalidades globales

* Gestion de usuarios
* Gestión de Instituciones
* Auditoría de acciones de los usuarios
* Gestión de los datos de la OEE

# Recursos necesarios

Para poder generar los reportes es necesario tener una fuente de letra específica la cual podemos con los siguientes comandos

Crear el directorio de destino:

|  |
| --- |
| sudo mkdir -p /usr/share/fonts/microsoft |

Copiar la fuente Arial.ttf del proyecto al directorio creado:

|  |
| --- |
| sudo cp Arial.ttf /usr/share/fonts/microsoft |

Actualizar el caché de fuentes:

|  |
| --- |
| sudo fc-cache -f -v |

Verificar si la fuente está instalada:

|  |
| --- |
| sudo fc-list |

# Despliegue backend

**Paso 1 - Instalar OpenJDK**

Verificar versión de Java, que coincida con la versión 11

**Paso 2 - Generar el ejecutable**

Se despliega la aplicación a través de perfiles de configuración para cada ambiente. Se definen 3 perfiles que hacen referencia al ambiente local, desarrollo (DEV) y producción (PROD) pudiendo agregarse mas, dependendiendo de la necesidad

Copiar el archivo application-example.properties que se encuentra en el directorio src/main/resources y renombrarlo como el environment seleccionado a levantar, ejemplo: application-local.properties. Modificar los valores por los del ambiente local.

Crear un archivo en el mismo directorio denominado application.properties cuyo contenido es el siguiente:

|  |
| --- |
| spring.profiles.active=@activatedProperties@ |

**Generar .jar**

Para generar el .jar con Maven, verificar si el archivo de configuración pom.xml contiene el siguiente fragmento de código

|  |
| --- |
| <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  <configuration>  <executable>true</executable>  </configuration> </plugin> |

Obs.: si no existe, agregar

Se agregó al pom.xml la dependencia URL Validator a la version 1.7, utilizado para validar la integridad de los datos recibidos.

Agregar la directiva

|  |
| --- |
| <!-- URL Validator--> <dependency>  <groupId>commons-validator</groupId>  <artifactId>commons-validator</artifactId>  <version>1.7</version> </dependency> |

Para generar el DDL de las tablas que se han agregado se debe configurar en el application-local.properties la seccion SPRING JPA las directivas:

|  |
| --- |
| spring.jpa.hibernate.ddl-auto=update spring.jpa.generate-ddl=true |

de esta manera se actualizará el esquema en la base de datos agregando los nuevos campos

luego de realizar la compilación revertir los cambios hechos a esas directivas

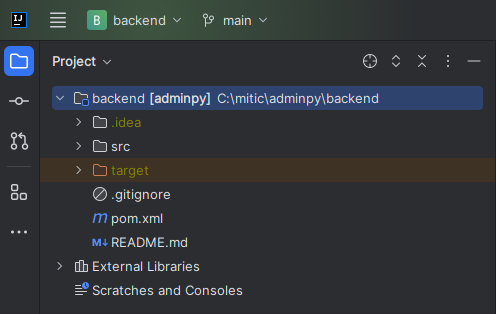
|  |
| --- |
| spring.jpa.hibernate.ddl-auto=none spring.jpa.generate-ddl=false |

Generar el ejecutable en src/main/resources

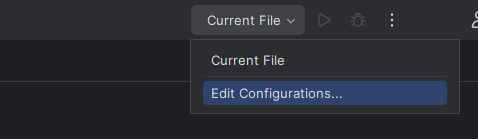
|  |
| --- |
| $ mvn clean package -p local $ mvn spring-boot:run -p local |

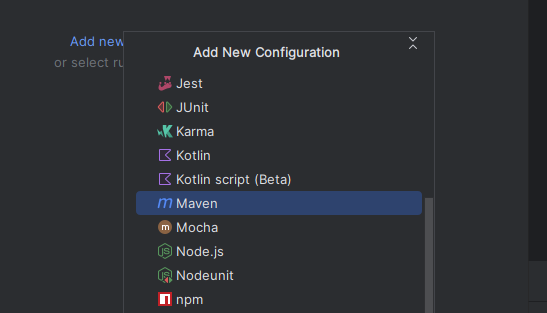
**Desplegar mediante IntelliJ**

1. Ubicar el proyecto en la carpeta backend

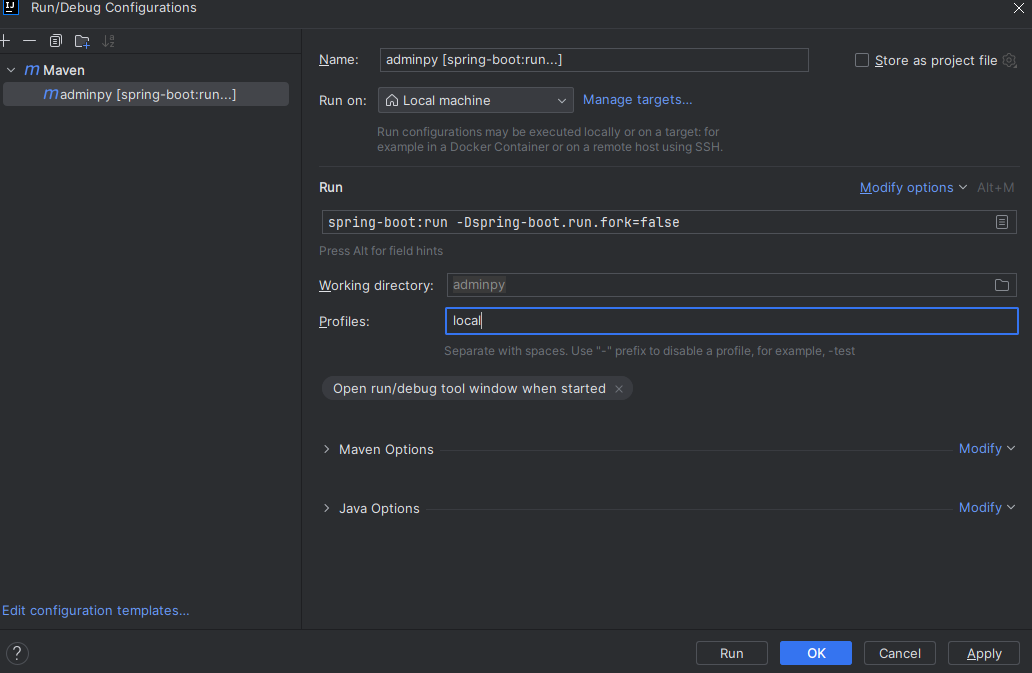


1. Agregar una nueva configuración para el despliegue del proyecto eligiendo Maven

****

****

1. Configurar con las siguientes directivas



|  |
| --- |
| spring-boot:run -Dspring-boot.run.fork=false |

5. Ejecutar el proyecto



**Paso 3 - Despliegue del ejecutable como un servicio (ambiente Linux)**

Se asume que la aplicación fue copiada en el directorio /opt/[directorio-aplicacion]/. Para instalar una aplicación Spring Boot como un servicio, se debe crear un enlace simbólico de la siguiente manera:

|  |
| --- |
| $ sudo ln -s /opt/[directorio-aplicacion]/[nombre-aplicacion].jar /etc/init.d/[enlace-aplicacion] |

# Ejemplo: sudo ln -s /opt/adminpy/backend-adminpy.jar /etc/init.d/adminpy

Una vez instalado, se puede iniciar y detener el servicio de la forma habitual. Por ejemplo, en un sistema basado en Debian, podría iniciarlo con el siguiente comando:

|  |
| --- |
| $ service [enlace-aplicacion] start |

# start, stop, restart, status

**Paso 4 - Restaurar permisos agregados en la base de datos**

Para restaurar los nuevos permisos agregados en la base de datos se debe copiar del archivo db/PermisosSQL.sql las sentencias INSERT, pegarlos en el administrador de la base de datos y ejecutarlos.

A continuación la tabla de permisos agregados\

|  |  |
| --- | --- |
| Permiso | Descripción |
| ver:menu:oee | Permite ver el agrupador o menú de Oee |
| ver:menu:administracion:planestic | Permite ver el menú de Planes TIC |
| planes:editar | Permite editar los Planes TIC |
| planes:borrar | Permite eliminar los Planes TIC |
| ver:menu:administracion:dependencias | Permite ver el menú de Dependencias |
| servicioOee:servicioInformacion:crear | Permite crear las informaciones de un servicio |
| servicioOee:servicioInformacion:borrar | Permite eliminar las informaciones de un servicio |
| organigrama:mostrarOrganigramaFuncionario | Mostrar Organigrama con funcionarios |
| auditoria:detalle | Permite ver el botón del detalle de auditoría |
| ver:menu:administracion:datosOee | Permite ver el menú de Datos OEE |
| usuarios:editarPerfil | Permite que el usuario pueda modificar su perfil |
| planes:crear | Permite crear Nuevos Planes TIC |
| planes:listar | Permite listar los Planes TIC |
| ver:menu:administracion:organigramatic | Permite crear nuevo dato de Organigrama |
| organigrama:crear | Permite listar los datos del Organigrama |
| organigrama:listar | Permite editar dato de Organigrama |
| organigrama:editar | Permite editar dato de Organigrama |
| PermisoAdmin | Permite ver todos los datos de la tabla sin importar la OEE |
| dependencia:downloadReport | Permite descargar Documento PDF de Dependencias |
| datosOee:descargar | Permite descargar Documento PDF de Datos OEE |
| oee:downloadReport' | Permite descargar Documento PDF de OEE |
| dependencia:pemisoLibre | Editar |
| organigrama:downloadReport | Permite descargar Documento PDF de Funcionarios (Organigrama) |
| organigrama:ver' | Permite visualizar el boton Ver Organigrama |
| datosOee:listar | Permite listar datos de la tabla Datos Oee |
| datosOee:editar | Permite editar Datos del Oee |
| dependencia:ver | Permite visualizar el boton Ver Organigrama |
| dependencia:listar | Permite listar los datos de las Dependencias |
| dependencia:crear | Permite crear nuevas Dependencias |
| dependencia:editar | Permite editar las Dependencias |
| dependencia:obtenerPorParametros | Permite obtener Dependencias por parametros |
| dependencia:getShowDependencia | Ver Organigrama Dependencia |
| oee:servicio:confirmarServicios | Permiso para confirmar servicios del OEE |

# 

# Despliegue Frontend

1. **Instalar nvm (Node Version Manager)**

nvm permite gestionar múltiples versiones de Node.js y npm de manera sencilla.

|  |
| --- |
| $ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash |

Este comando descargará e instalará nvm en tu sistema.

Cierra y abre la terminal.

Instalar Node.js y npm con nvm

|  |
| --- |
| $ nvm install 16.17.0 |

Esto instalará Node.js en la versión 16.17.0 y también configurará la versión adecuada de npm para esa versión de Node.js.

1. **Configurar proxy local**

Copiar el archivo /frontend/proxy.config.example.json y renombrarlo a proxy.config.json. Modificar los valores del target para acceso al backend.

|  |
| --- |
| {  "/api/\*": {  "target": "http://{IP or LOCALHOST}:{PORTH}/",  "secure": false,  "loglevel": "debug",  "changeOrigin": true  } } |

1. **Instalación de CLI de angular**

|  |
| --- |
| $ cd /frontend $ npm install -g @angular/cli |

1. **Comando sass para compilar css.**

|  |
| --- |
| $ npm install -g node-sass $ node-sass src/assets/layout/css/layout-light.scss src/assets/layout/css/layout-light.css $ node-sass src/assets/theme/indigo/theme-light.scss src/assets/theme/indigo/theme-light.css |

1. **Instalar paquetes y librerías del package.json**

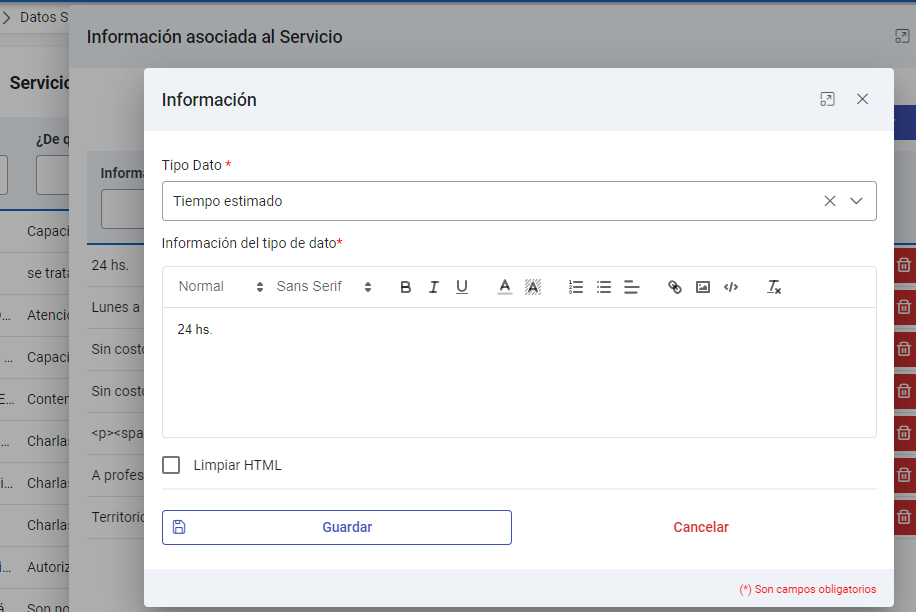
|  |
| --- |
| $ npm install |

En caso de presentar error en la instalación de los paquetes lanzar el comando con –force

1. **Se agregaron 2 nuevas librerías**

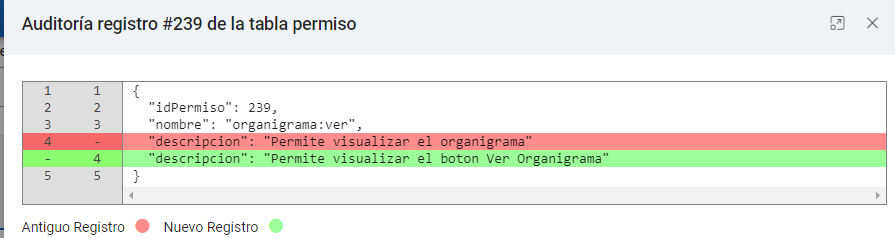
**quill:** editor WYSIWYG. Utilizado en la pantalla de servicios, para agregar o editar la información asociada al servicio

|  |
| --- |
| npm install quill |



**diff:** componente de Angular para comparar diferencias en el texto. Utilizado para presentar las diferencias en Auditoría para resaltar lo modificado en la acción de modificar

|  |
| --- |
| npm install ngx-diff@1.0.0 diff-match-patch-ts --save --force |



Estos ya están incluidos en el archivo package.json

1. **Levantar instancia de desarrollo**

|  |
| --- |
| $ npm start |

1. **Compilar el proyecto frontend para ambiente producción.**

|  |
| --- |
| $ npm run build --output-hashing=all |

La opción output-hashing=all borra/renueva el caché del proyecto

# Diccionario de Datos

Tabla **Planes TIC (plan)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Dominio** | **Tipo de Campo** | **Indicación** | **Validación** |
| **id\_plan** | Clave principal | Integer | Numérico | Entrada obligatoria y única |
| **anho** | Año del plan | Varchar | Alfanumérico |  |
| **fecha** | Fecha del plan | Date | Fecha |  |
| **link\_plan** | URL del plan | Varchar | Alfanumérico |  |
| **cantidad\_funcionarios\_tic** | Cantidad de los funcionarios TIC | Varchar | Alfanumérico |  |
| **cantidad\_funcionarios\_admin** | Cantidad de los funcionarios de Administración | Varchar | Alfanumérico |  |
| presupuesto\_tic\_anual | Presupuesto anual TIC | Varchar | Alfanumérico |  |
| **estado** | Estado del registro | Boolean | Boolean |  |

Tabla **Dependencia (dependencia)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Dominio** | **Tipo de Campo** | **Indicación** | **Validación** |
| **id\_dependencia** | Clave principal | Integrer | Numérico | Entrada obligatoria y única |
| **descripcion\_dependencia** | Nombre de la Dependencia | Varchar | Alfanumérico |  |
| **estado\_dependencia** | Estado actual de la dependencia | Boolean | Boolean |  |
| **fecha\_creacion** | Fecha en la que se creó la Dependencia | Timestamp | Marca de tiempo |  |
| **orden** | Orden en el cual aparecen | Integer | Numérico |  |
| **id\_dependencia\_padre** | Clave de la Dependencia de la cual depende esta Dependencia | Integer | Numérico |  |
| **id\_oee** | Clave principal de la OEE | Integer | Numérico |  |
| **codigo** | Código de la Dependencia | Varchar | Alfanumérico |  |

Tabla Funcionario (**funcionario**)

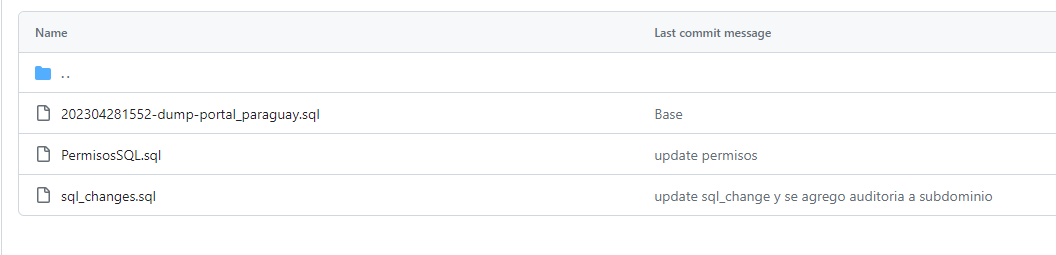
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Dominio** | **Tipo de Campo** | **Indicación** | **Validación** |
| **id\_funcionario** | Clave principal | Integer | Numérico | Entrada obligatoria y única |
| **apellido** | Apellido del funcionario | Varchar | Alfanumérico |  |
| **cargo** | Cargo del Funcionario | Varchar | Alfanumérico |  |
| **cedula** | Cédula del Funcionario | Varchar | Alfanumérico |  |
| **correo\_institucional** | Correo del Funcionario | Varchar | Alfanumérico |  |
| **estado\_funcionario** | Estado actual del Funcionario | Boolean | Boolean |  |
| **fecha\_creacion** | Fecha en la que se creó el registro | Timestamp | Marca de tiempo |  |
| **nombre** | Nombre del Funcionario | Varchar | Alfanumérico |  |
| **id\_dependencia** | Clave principal de la Dependencia a la cual pertenece | Integer | Numérico |  |

# Modificaciones de Base de Datos

El archivo con las modificaciones se encuentra en la ruta db/sql\_changes.sql

Se debe ejecutar los ALTER en el administrador de Base de Datos teniendo en cuenta el seguimiento de las fechas especificadas en cada uno. (Se adjunta fecha y hora del cambio realizado en el archivo)

**Carpeta bd:**



**Listado de cambios realizados:**

