

SISTEMA de GESTIÓN DOCUMENTAL Y CONTROL de CORRESPONDENCIA

DETALLES DEL SISTEMA

Este sistema cuenta con 5 módulos:

1. Módulo de gestión de usuarios
2. Módulo de gestión de requisitos
3. Módulo de gestión de tipo de trámite
4. Módulo de gestión y derivación de trámites
5. Módulo de gestión documental para documentos digitales

Los usuarios tienen acceso a una o más funciones de los módulos listados anteriormente de acuerdo al rol asignado, para ello se especifican 4 roles predefinidos:

1. El rol de ADMINISTRADOR cuenta con permisos para:
 - Crear Usuarios
 - Editar Usuarios
 - Desactivar Usuarios
 - Reactivar Usuarios
 - Crear Requisito
 - Editar Requisito
 - Eliminar Requisito (solo si no forma parte de un tipo de trámite)
 - Crear Tipos de Trámites
 - Editar Tipos de Trámites
 - Eliminar Tipos de Trámites (solo si no existen trámites creados del tipo en cuestión)
2. El rol de SECRETARÍA cuenta con permisos para:
 - Crear Trámites
 - Editar Trámites (recién creados y sin derivar)
 - Eliminar Trámites (recién creados y sin derivar)
 - Derivar Trámite (hacia secciones predefinidas)
 - Adjuntar Archivos (solo si el trámite se encuentra en su bandeja y no existe otro archivo con el mismo nombre asociado al trámite)
 - Archivar Trámite
3. El rol de VERIFICADOR cuenta con permisos para:
 - Derivar Trámites (hacia secciones predefinidas)
 - Archivar Trámites
 - Adjuntar Archivos (solo si el trámite se encuentra en su bandeja y no existe otro archivo con el mismo nombre asociado al trámite)

El encargado de administrar usuarios, para el caso sea el usuario que cuenta con el rol ADMINISTRADOR, es el que cuenta con la facultad de definir la sección para cada usuario usuario por medio. A cada sección se le asignó un rol predefinido:

1. La sección ADMINISTRADOR tiene asociado el rol ADMINISTRADOR
2. La sección PRESIDENCIA tiene asociado el rol SECRETARÍA
3. La sección VICE PRESIDENCIA tiene asociado el rol SECRETARÍA
4. La sección STRIA. RÉGIMEN INTERNO Y PERSONAL tiene asociado el rol SECRETARÍA
5. La sección SECRETARÍA HACIENDA tiene asociado el rol SECRETARÍA
6. La sección SECRETARÍA GENERAL tiene asociado el rol SECRETARÍA
7. La sección STRIA. BIENESTAR SOCIAL Y VIVIENDA tiene asociado el rol SECRETARÍA
8. La sección STRIA. DE EDUCACIÓN Y CULTURA tiene asociado el rol SECRETARÍA
9. La sección STRIA. DE ACTAS Y DEPORTE tiene asociado el rol SECRETARÍA
10. La sección DIRECTOR ADMINISTRATIVO tiene asociado el rol SECRETARÍA
11. La sección UNIDAD DE SISTEMAS tiene asociado el rol SECRETARÍA
12. La sección UNIDAD DE ASESORÍA JURÍDICA tiene asociado el rol SECRETARÍA
13. La sección UNIDAD DE AUDITORÍA tiene asociado el rol SECRETARÍA
14. La sección RESPONSABLE DE PRESTACIONES tiene asociado el rol SECRETARÍA
15. La sección ATENCIÓN EN PLATAFORMA tiene asociado el rol VERIFICADOR
16. La sección TESORERÍA tiene asociado el rol SECRETARÍA
17. La sección STRIA. RELACIONES PUBLICAS, PRENSA Y PROPAGANDA tiene asociado el rol SECRETARÍA

CONSIDERACIONES DE USO

- Cuando se crean usuarios el sistema asigna un nombre de usuario parcial que se valida al momento de enviar el formulario para evitar duplicidad de nombres de usuarios, sin embargo este campo es editable.
- La primera contraseña de cada usuario es el dato del documento de identidad (con el Complemento Alfanumérico en Mayúsculas), siendo posible el cambio de contraseña por cada usuario en la vista de PERFIL o por medio de un usuario administrador en el formulario de edición de usuarios.
- Cuando se crean hojas de ruta el sistema asigna una hoja de ruta parcial que se valida al momento de enviar el formulario para evitar duplicidad de códigos de hojas de ruta, sin embargo este campo es editable.
- Los contadores que el sistema utiliza para generar códigos de hojas de ruta se reinician anualmente siempre y cuando el servidor tenga habilitada las tareas programadas de Cron y se hayan seguido los pasos indicados en la [documentación de Laravel](#).

REQUISITOS DEL SERVIDOR

- Se debe disponer de uno o más servidores con las siguientes aplicaciones instaladas y funcionales:
 1. PHP versión 7.4.22
 2. Apache versión 2.4.48 o Nginx versión 1.19.6
 3. MariaDB versión 10.4.20
 4. OpenSSL versión 1.1.1
 5. NodeJS versión 14.17.3 o superior
 6. Yarn versión 1.22.11 o superior
 7. MQTT Mosquitto Broker versión 2.0.12

PASOS PARA LA INSTALACIÓN

Para seguir los pasos listados continuación se debe tener abierta una consola con la ruta ubicada en la raíz del de la carpeta del proyecto.

1. Clonar el código fuente:

```
$ git clone --recurse-submodules --remote-submodules https://github.com/djimenezjerez/correspondencia.git
```

2. Establecer las variables de entorno, renombrar **.env.example** a **.env** y edita los datos de acuerdo a los datos extraídos del servidor donde funcionará el sistema:

```
$ copy .env.example .env
```

3. Establecer datos del primer usuario administrador:

```
$ copy admin_data.json.example storage/app/admin_data.json
```

4. Editar los datos para el usuario administrador en el archivo recién copiado a **storage/app/admin_data.json**.
5. Editar el dominio en el archivo **config/sanctum.php**, dentro de la variable *stateful*, reemplazar *localhost* por el dominio especificado por el servidor.
6. Instalar las dependencias:

```
$ composer install
$ php artisan key:generate
$ yarn install
$ php artisan migrate:install
$ php artisan migrate
```

7. Compilar el código fuente del frontend:


```
$ yarn prod
```

8. Cargar datos iniciales:

```
$ php artisan db:seed --class=DatabaseSeeder
```

9. Habilitar el cargado de imágenes dentro de PDF:

```
$ sed -i.bak 's/$alpha_channel = $imagickClonable ? clone $imagick : $imagick->clone();/$alpha_channel = clone $imagick;/g' vendor/c
```



Concluyendo estos pasos, se puede ingresar al navegador y apuntar la URL a la dirección del dominio del servidor donde se instaló el sistema.

PASOS PARA LA ACTUALIZACIÓN

Para seguir los pasos listados continuación se debe tener abierta una consola con la ruta ubicada en la raíz del de la carpeta del proyecto.

1. Actualizar las variables de entorno del archivo **.env** de acuerdo a los datos del ejemplo del archivo **.env.example**
2. Instalar las nuevas dependencias:

```
$ composer install
$ yarn install
$ php artisan migrate
```

3. Compilar el código fuente del frontend:

```
$ yarn prod
```

4. Cargar los nuevos datos:

```
$ php artisan db:seed --class=DatabaseSeeder
```

5. Habilitar el cargado de imágenes dentro de PDF:

```
$ copy /Y docker/Cpdf.php vendor/dompdf/dompdf/lib/Cpdf.php
```

Concluyendo estos pasos, se puede ingresar al navegador y apuntar la URL a la dirección del dominio del servidor donde se instaló el sistema.

PUESTA EN PRODUCCIÓN MEDIANTE DOCKER

- Para levantar el sistema mediante virtualización de contenedores se necesitan los siguientes requisitos:
 1. Docker versión 20.10.7 o superior
 2. Docker Compose versión 1.29.2 o superior

Los pasos a seguir para levantar los servicios necesarios son los siguientes:

1. Clonar el código fuente:

```
$ git clone --recurse-submodules --remote-submodules https://github.com/djimenezjerez/correspondencia.git
```

2. Editar el archivo de variables de entorno **docker/.env.example** y de acuerdo a ello modificar el dominio en el archivo **config/sanctum.php**, dentro de la variable *stateful*, reemplazar *localhost* por el dominio especificado para el servidor Nginx o Apache.
3. Copiar el archivo de variables de entorno editado anteriormente al directorio laradock:

```
$ copy /Y docker/.env.example laradock/.env
```

4. Copiar los archivos necesarios para levantar correctamente los contenedores:

```
$ copy /Y docker/docker-compose.yml laradock/docker-compose.yml
$ copy /Y docker/mosquitto.conf laradock/mosquitto
```

5. Descargar las imágenes de los contenedores:

```
$ cd laradock
$ docker-compose build --parallel nginx php-fpm workspace mariadb mosquitto
```

6. Levantar el sistema:

```
$ docker-compose up -d nginx php-fpm workspace mariadb mosquitto
```

7. Verificar que todos los contenedores se encuentren en ejecución **State = Up**

```
$ docker-compose ps
```

Con ello se pueden seguir los [pasos de instalación](#) o [pasos para la actualización](#) mencionados anteriormente en el presente manual, tomando muy en cuenta que cada comando de **composer**, **php** o **yarn** debe ser ejecutado desde el directorio de *laradock* y precedido por el comando **docker-compose exec --user laradock workspace COMANDO**(composer, php o yarn).