



Plataforma de Seguimiento de Casos COVID-19 Paraguay

Manual de Instalación



TABLA DE CONTENIDO

TABLA DE CONTENIDO	2
1. PROYECTO	3
2. INTRODUCCIÓN	3
3. VISIÓN GENERAL.....	3
4. PROCEDIMIENTO INSTALACIÓN – VISIÓN GENERAL.....	4
5. PROCEDIMIENTO DE INSTALACIÓN	5
5.1. ACTIVIDAD 1: VALIDAR PRERREQUISITOS	5
5.1.1. ESPECIFICACIONES TÉCNICAS.....	5
5.1.2. PERMISIONES Y AMBIENTE.....	5
5.1.3. ACCESO Y DESCARGA AL REPOSITORIO:.....	5
5.2. ACTIVIDAD 2: REALIZACIÓN COPIA DE SEGURIDAD	6
5.3. ACTIVIDAD 3: PROCEDIMIENTO DE AJUSTES EN BASE DE DATOS	6
5.4. ACTIVIDAD 4: PROCEDIMIENTO DE INSTALACIÓN EN EL SERVIDOR.....	7
LEVANTAR EL PROYECTO BACKEND	7
LEVANTAR EL PROYECTO FRONTEND	7
OBTENER SERVIDOR DE APLICACIÓN.....	8
CONFIGURAR MÓDULO DE POSTGRESQL EN WILDFLY	8
CONFIGURACIÓN DEL DATASOURCE EN EL STANDALONE DEL WILDFLY	8
5.5. ACTIVIDAD 5: CONFIGURACIÓN DE LAS TAREAS AGENDADAS.....	9
5.5.1. DESCRIPCIÓN TAREAS BATCH – SISTEMA SEGUIMIENTO PACIENTES	9
5.5.2. INSTALACIÓN TAREAS BATCH – SISTEMA SEGUIMIENTO PACIENTES.....	10
5.6. ACTIVIDAD 6: PRUEBAS DE VERIFICACIÓN Y VALIDACIÓN	11
6. ENVÍO DE MENSAJES PUSH A UN DISPOSITIVO MÓVIL UTILIZANDO FIREBASE.....	11
7. FLUJO DE EXCEPCIÓN	11
C. ACTIVIDAD 3.1 PROCEDIMIENTO ROLLBACK BASE DE DATOS	11
D. ACTIVIDAD 4.1 PROCEDIMIENTO ROLLBACK SERVIDOR DE APLICACIONES	12

1. Proyecto

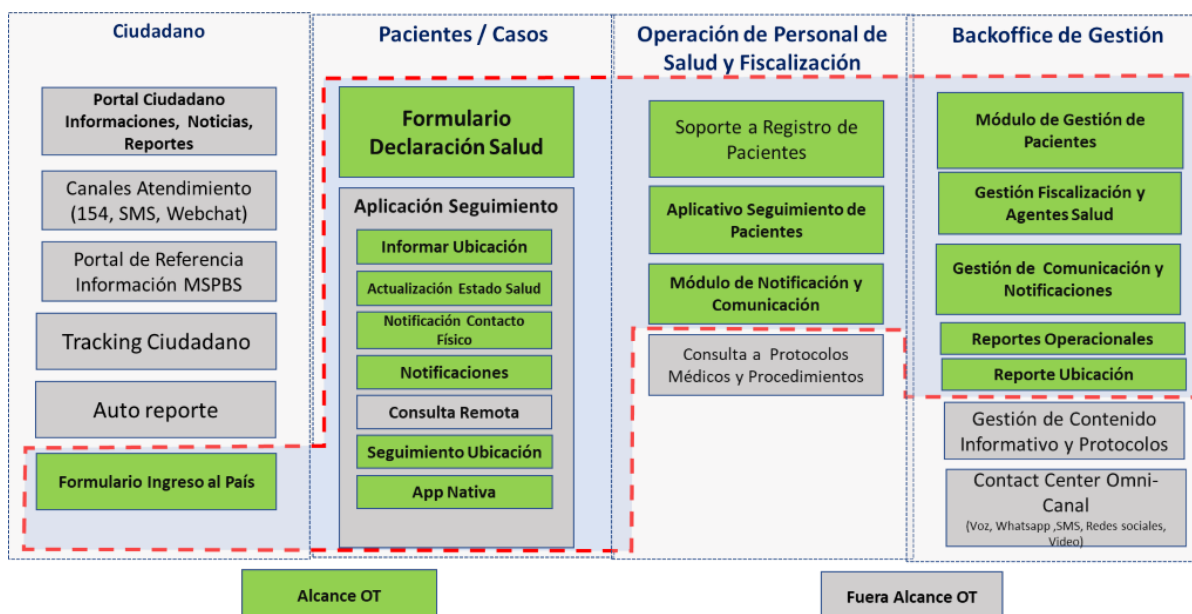
Plataforma de seguimiento de pacientes en cuarentena, a causa del coronavirus (COVID-19) en Paraguay.

2. Introducción

El presente documento está dirigido al personal técnico responsable de instalar y configurar la Plataforma de Seguimiento de Casos de Covid 19; por tanto se asume que el lector está familiarizado con conceptos básicos de administración bajo el sistema operativo Linux y servidor Nginx.

3. Visión General

A continuación, se presenta una visión general de los principales componentes de la plataforma cuyo alcance es objeto del presente documento. Los módulos y funcionalidades delimitados por la línea puntillada roja son los módulos contemplados en el presente documento.



El mapeo entre Módulo y Proyectos es el siguiente:

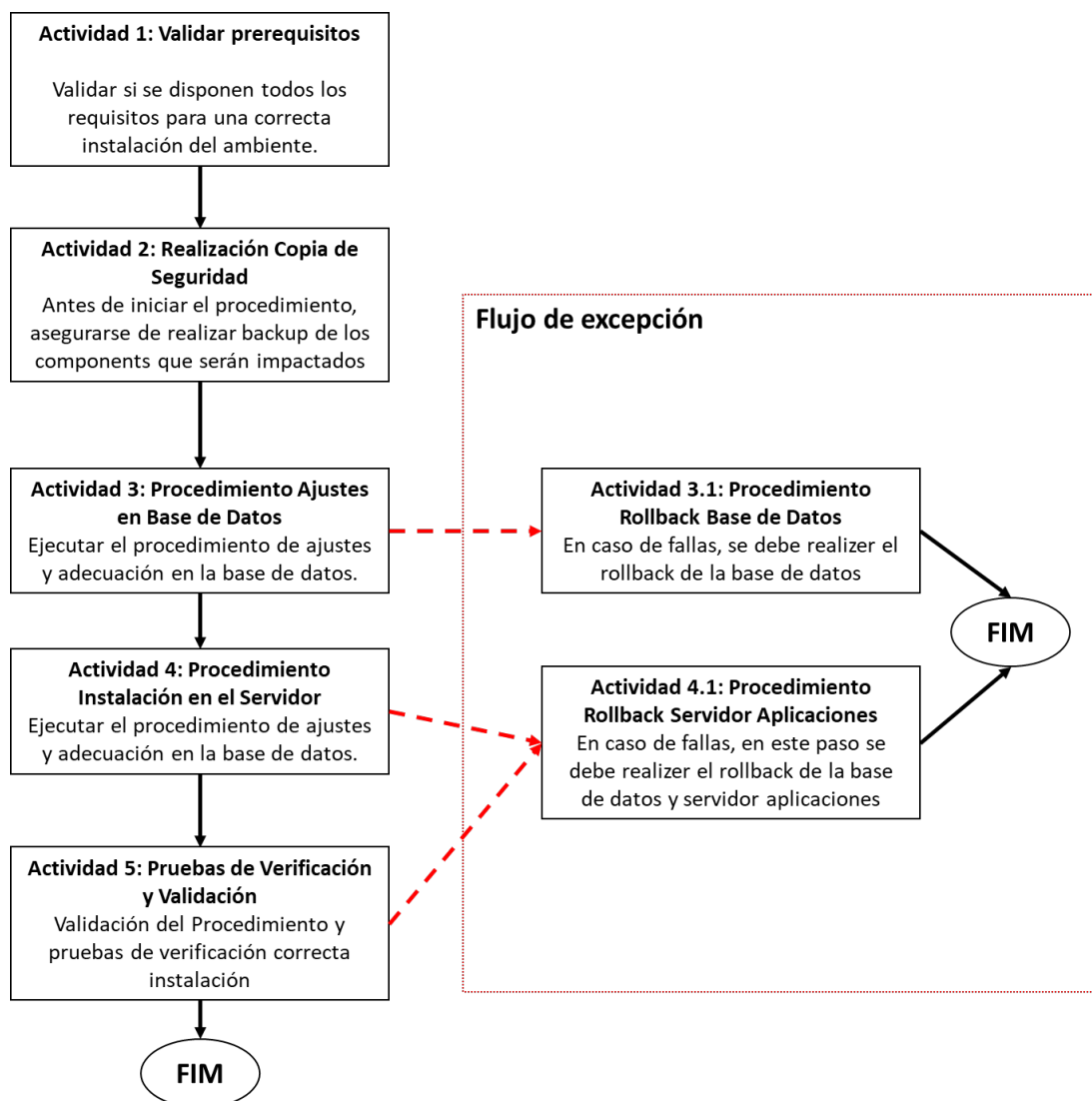
Módulo	Proyecto Asociado	Tecnología
Ciudadano – Formulario Ingreso al País Formulario Declaración Salud	portal-covid-core	EJB, JPA, Hibernate, Jax-RS, Jackson
Pacientes – Casos – Webapp (Informar Ubicación, Actualizar Estado Salud, Notificación Contacto Físico)	portal-covid-wpa	Vue.js, Bootstrap, Awesome
Pacientes / Casos – App Nativa	portal-covid-app	Flutte, Firebase



Backoffice de Gestión	portal-covid-frontend	Angular 7, Primeng, Bootstrap, Font Awesome
-----------------------	-----------------------	---

4. Procedimiento Instalación – Visión General

Para una correcta Instalación de la presente versión, se presenta una visión general de los pasos que se deben ejecutar:



IMPORTANTE: debido a que el procedimiento posee ajustes a nivel de servidor de aplicaciones y base de datos, en caso de ejecutar el presente procedimiento en producción, se recomienda la puesta del ambiente en modo de mantenimiento.



5. Procedimiento de Instalación

5.1. Actividad 1: Validar Pre-requisitos

5.1.1. Especificaciones Técnicas

Requisitos de Software

Servidor de Aplicación: Wildfly 18.0.1
Sistema Operativo: Centos 7.
Nginx

Lenguaje de Programación

Backend: Java EE
Frontend: Javascript. Framework Angular 7

Base de Datos.

Postgresql 12.2

Otras Tecnologías

Maven: Gestión de dependencias para proyecto Backend. Versión 3.3.9
Git: control de versionamiento.
Angular CLI 7.1.2: Scaffolding para estructura base del proyecto.
Node.JS: 11.10.0

5.1.2. Permisos y Ambiente

Se deberá tener acceso a los siguientes ambientes y perfiles de usuario:

- Usuario dueño de la base de datos, para poder modificar la estructura de las tablas existentes y poder crear nuevas tablas.
- Usuario del Gitlab de MITIC para poder descargar los proyectos.
- Usuario del servidor para poder compilar los proyectos y ejecutar los servicios (wildfly, Postgresql, MongoDB, Nginx o Apache).

5.1.3. Acceso y Descarga al Repositorio:

Se deberán descargar los proyectos de El Portal Paraguay que se encuentran en el GIT. El portal Paraguay se encuentra dividido de la siguiente manera:

Proyecto:	covidpy-frontend-operator
Descripción:	Es el proyecto angular frontend de los operadores de salud
URL Git:	https://github.com/covidpy/covidpy-frontend-operator



Clonar:	<code>git clone git@github.com:covidpy/covidpy-frontend-operador.git</code>
Rama:	Mudarse a la rama de trabajo <code>git checkout rama_trabajo</code>

Proyecto:	<code>covidpy-core</code>
Descripción:	Es el proyecto backend encargado de realizar las consultas a la base de datos y lógica de la aplicación.
URL Git:	<code>https://github.com/covidpy/covidpy-core</code>
Clonar:	<code>git clone git@github.com:covidpy/covidpy-core.git</code>
Rama:	Mudarse a la rama de trabajo <code>git checkout rama_trabajo</code>

Proyecto:	<code>covidpy-frontend-ciudadano</code>
Descripción:	Es el proyecto frontend de los pacientes
URL Git:	<code>https://github.com/covidpy/covidpy-frontend-ciudadano</code>
Clonar:	<code>git clone git@github.com:covidpy/covidpy-frontend-ciudadano.git</code>
Rama:	Mudarse a la rama de trabajo <code>git checkout rama_trabajo</code>

Importante: el procedimiento de “merge” entre el BRANCH utilizado y BRANCH developer del GIT deberá ser realizado por el administrador del repositorio.

5.2. Actividad 2: Realización Copia de Seguridad

Para poder realizar un backup de la base de datos Postgresql se puede usar la herramienta `pg_dump`. Ejemplo

```
pg_dump dbname> db.sql
```

También realizar un backup de los war del servidor wildfly copiando el archivo `portal-covid-core.war` de la carpeta `standalone/deployments` del wildfly.

Para realizar backup del frontend copiar todos los archivos del document root de Nginx o Apache que esté utilizando.

5.3. Actividad 3: Procedimiento de Ajustes en Base de Datos

1. Conectarse al postgres con el usuario dueño de la base de datos.
2. Ejecutar los scripts *que se encuentran en la carpeta `portal-covid-core\portal-covid-core\src\main\resources\db\migration` en orden ascendente.*



5.4. Actividad 4: Procedimiento de Instalación en el Servidor

Levantar el proyecto backend

Utilizando Eclipse posicionarse sobre el proyecto y con el botón derecho del mouse buscar la opción **“Export”** luego **“War File”**.

Elegir en que carpeta desea crear el archivo war y seleccione la opción **“Overwrite existing file”**. Por último clic en el botón **Finish**.

Proceda a deployar en el servidor wildfly.

Levantar el proyecto frontend

La versiones de Node en las que se probó compilar el frontend fueron la 12 y la 11. En la versión 12 no pudo compilarse el proyecto exitosamente. Si se pudo compilar exitosamente en la versión 11.

Para instalar Node versión 11 en Centos se debe ejecutar el siguiente comando para agregar el repositorio de Node 11

```
$ sudo curl -sL https://rpm.nodesource.com/setup_11.x | bash -
```

```
$ sudo yum install nodejs
```

Para compilar el proyecto del operador de salud se debe ejecutar el siguiente comando:

```
$ ng build --prod --build-optimizer
```

Este comando generará una carpeta /dist, el contenido de la misma debe ser copiada en el servidor de nginx directorio /var/www/.

Para compilar el proyecto del paciente se debe ejecutar el siguiente comando:

```
$ npm run build
```

Este comando generará una carpeta /dist, el contenido de la misma debe ser copiada en el servidor de nginx directorio /var/www/.



Obtener Servidor de aplicación

Descargue el servidor desde la página oficial de wildfly con la versión 18.0.1:

<http://wildfly.org/downloads/>

Configurar Módulo de PostgreSQL en Wildfly

1) Abrir la carpeta del directorio

\$WILDFLY_HOME/modules/system/layers/base/org/postgresql/main

2) Crear las carpetas: org, postgresql y main si no se encuentran creadas.

3) Copiar el driver del postgresql y pegarlo en la carpeta main, que fue creada en el punto 1

4) Dentro de la carpeta main, crear un archivo de nombre module.xml y agregue el siguiente contenido:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="org.postgresql">
```

```
  <resources>
    <resource-root path="postgresql-9.3.1208.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Configuración del datasource en el standalone del Wildfly

1. Abrir la carpeta del directorio \$WILDFLY_HOME/standalone/configuration/
2. Dentro de la carpeta /configurations/, abra el archivo standalone-full.xml o standalone.xml y agregue el driver de postgresql que está en negrita (líneas 5,6,7) del ejemplo de abajo:

```
<drivers>
  <driver name="h2" module="com.h2database.h2">
    <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
  </driver>
  <driver name="postgresql" module="org.postgresql">
    <driver-class>org.postgresql.Driver</driver-class>
  </driver>
</drivers>
```

3. Dentro de la carpeta /configurations/, abra el archivo standalone-full.xml o standalone.xml busque los marcadores <datasources></datasources> y dentro de los mismos agregue el siguiente contenido:

```
<datasource jndi-name="java:jboss/datasources/covid19DS" pool-name="covid19DS" enabled="true"
use-java-context="true">

  <connection-url>jdbc:postgresql://localhost:5432/covid19</connection-url>
  <driver-class>org.postgresql.Driver</driver-class>
```




```
<driver>postgresql-9.3.1208.jar</driver>
<transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
<pool>
  <min-pool-size>10</min-pool-size>
  <max-pool-size>100</max-pool-size>
  <prefill>true</prefill>
  <use-strict-min>false</use-strict-min>
  <flush-strategy>FailingConnectionOnly</flush-strategy>
</pool>
<security>
  <user-name>postgres</user-name>
  <password>postgres</password>
</security>
<statement>
  <prepared-statement-cache-size>32</prepared-statement-cache-size>
</statement>
</datasource>
```

5.5. Actividad 5: Configuración de las tareas agendadas

El proyecto posee un conjunto de tareas en batch que deben ser configuradas para el envío de notificaciones a los usuarios. A continuación una descripción de las tareas batch existentes.

5.5.1. Descripción Tareas Batch – Sistema Seguimiento Pacientes

Nombre Tarea	Descripción Tarea	Periodicidad	Comando a incluir en el crontab	Usuario Ejecución	Marcador de Log (para hacer grep)
Agendados	Loguearse como el usuario agendamiento, que tiene el permiso para invocar al proceso de agendamiento. Ejecutar el proceso de agendamiento con las credenciales obtenidas en	De 8 a 23hs, intervalo de 2 horas	/home/wildfly/covid/scripts/agendados.bash	wildfly	agendados-



	el paso anterior y enviar los sms				
pacientesSinRegistroConfirmado12	Notificación a los pacientes positivos que no se registraron pasadas las 12 horas y envío de sms	De 8 a 23hs, intervalo de 2 horas	/home/wildfly/covid/scripts/pacientesSinRegistroConfirmado12.bash	wildfly	pacientesSinRegistroConfirmado12-
pacientesSinRegistroUbicacionConfirmado3	Notificación a los pacientes confirmados que no reportaron su ubicación pasadas las 3 horas de retraso y envío de sms	De 8 a 23hs, intervalo de 2 horas	/home/wildfly/covid/scripts/pacientesSinRegistroUbicacionConfirmado3.bash	wildfly	pacientesSinRegistroUbicacionConfirmado3-
pacientesSinReporteSalud	Notificación a los pacientes positivos que no reportaron su estado de salud con 12 horas de retraso y envío de sms	De 8 a 23hs, intervalo de 2 horas	/home/wildfly/covid/scripts/pacientesSinReporteSalud.bash	wildfly	pacientesSinReporteSalud-
pacientesSinRegistroSospechoso	Notificación a los pacientes sospechosos que no se registraron pasadas las 24 horas de retraso y envío de sms	No ejecutar a pedido del MSPyBS.			

5.5.2. Instalación Tareas Batch – Sistema Seguimiento Pacientes

1. Loguearse como usuario wildfly
2. Copiar los archivos a /home/wildfly/covid/scripts
3. Instalar en el crontab conforme abajo
 - a. Ejecutar como usuario wildfly el comando crontab -e
 - b. Agregar las siguientes líneas en el crontab del usuario:

```
0 8-22/2 * * * /home/wildfly/covid/scripts/agendados.bash
15 8-22/2 * * * /home/wildfly/covid/scripts/pacientesSinRegistroConfirmado12.bash
30 8-22/2 * * * /home/wildfly/covid/scripts/pacientesSinRegistroUbicacionConfirmado3.bash
45 8-22/2 * * * /home/wildfly/covid/scripts/pacientesSinReporteSalud.bash
```

Nota: si se desear suspender el agendamiento basta con comentar la línea en el crontab



5.6. Actividad 6: Pruebas de Verificación y Validación

Setear el rol “Operador” a un usuario del sistema, loguearse al sistema con ese usuario e ingresar al sistema.

6. Envío de mensajes Push a un dispositivo móvil utilizando Firebase

Cuando un paciente se loguea en la aplicación nativa en el registro de la tabla covid19admin.usuario de ese usuario, en la columna fcm_registration_token se registrará el token de Firebase registrado para la aplicación.

Utilizando dicho token se puede enviar una notificación push a ese usuario de la siguiente manera:

```
curl --location --request POST 'https://fcm.googleapis.com/fcm/send' \  
--header 'Accept: application/json' \  
--header 'Authorization: key= AQUÍ_COMPLETAR_KEY_SERVER' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "to": "FCM_REGISTRATION_TOKEN_USUARIO",  
  "collapse_key": "type_a",  
  "notification": {  
    "title": "Examen Laboratorial",  
    "body": "NOMBRE_PACIENTE, tiene examen laboratorial en fecha  
11/07/2020. Local: San Lorenzo. Tiene autorizacion de circulacion para tal efecto. MSPBS.",  
    "sound": "default"  
  },  
  "data": {  
    "key_1": "Value for key_1",  
    "key_2": "Value for key_2"  
  }  
}'
```

7. Flujo de Excepción

c. Actividad 3.1 Procedimiento Rollback Base de Datos

Restaurar el backup realizado en el paso 1 con el siguiente comando:

```
psql -d newdb -f db.sql
```



d. Actividad 4.1 Procedimiento Rollback Servidor de Aplicaciones

Para restaurar el backup del servidor de aplicaciones reemplazar el archivo portal-covid-core.war en la carpeta standalone/deployments del wildfly.

Para restaurar el backup del frontend copiar todos los archivos del backup del document root del Nginx o Apache al directorio del servidor.