

## Precisiones para el despliegue de Aplicación ASI réplica

### Recursos necesarios:

1. Conexión a internet mediante un router casero
2. Android Studio



3. Spring Tool Suite



4. pgAdmin



### Precisiones:

i) Se debe instalar el software de PostgreSQL en el sistema operativo (se utilizó Windows 10 en este proyecto). Crear un usuario llamado “dpppt” y contraseña “dpppt”. Por otro lado, se debe descargar el archivo **dpppt.sql** que se encuentra en el repositorio y ejecutarlo en pgAdmin para crear la base de datos.

master [Aplicaciones-moviles-para-contact-tracing-usando-bluetooth](#) / [Testing caja blanca](#) / Código Base de Datos del Backend / Go to file Add file ...

VictorJEM Carga del Backend c32490 1 hour ago History

..

Backend Relational Tables.png	Carga del Backend	1 hour ago
README.txt	Carga del Backend	1 hour ago
dpppt.sql	Carga del Backend	1 hour ago

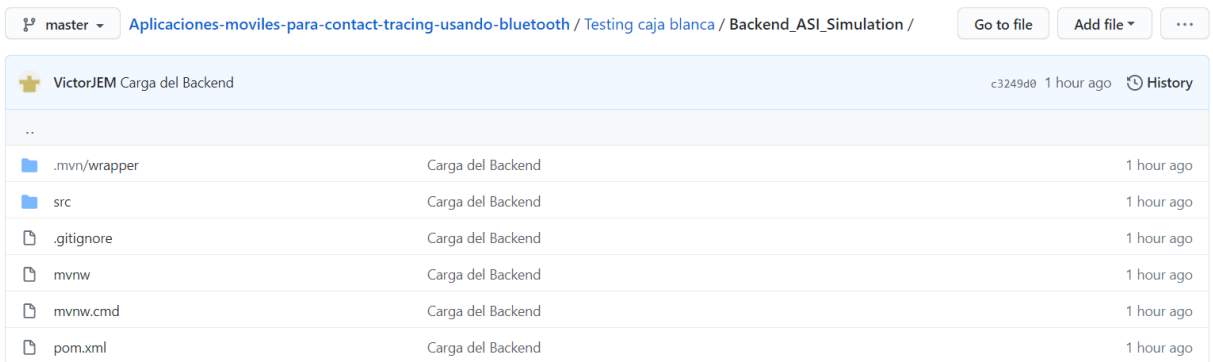
README.txt

→ La tabla auth\_codes almacena los código COVID que autoriza a los usuarios para que se registren como infectados por el virus.

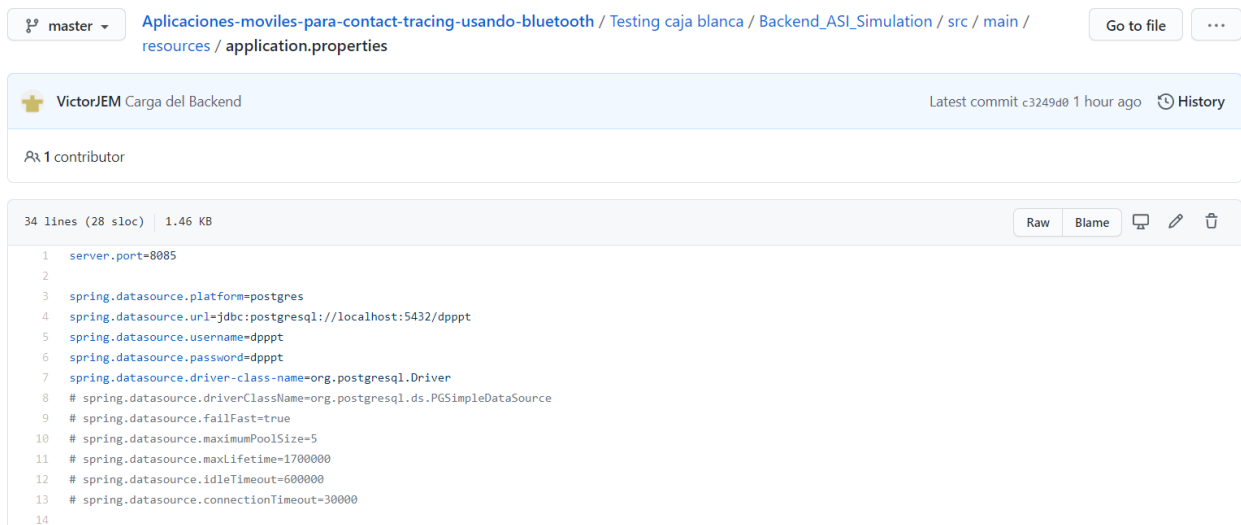
→ La tabla exposees contiene la información de los usuarios infectados que será descargada por la aplicación para comparar y alertar de posible contagio.

ii) Teniendo instalado el programa Spring Tool Suite, se debe descargar la carpeta Backend\_ASI\_Simulation que contiene el código que interactúa con los usuarios y la base de datos.

Ya en el programa se puede abrir el proyecto ejecutando únicamente el archivo **pom.xml** que se observa en la imagen de abajo.




Por defecto, el servidor queda asociado con la dirección IP privada que utiliza la PC conectada a la red local del router casero. Pero es posible cambiar el puerto (server.port) que escuchará las solicitudes mediante el archivo **application.properties** como se observa en la imagen de abajo.



Con lo anterior, ya se puede encender el servidor desde el programa Spring Tool Suite.

iii) Una vez instalado Android Studio, se debe descargar la carpeta ASI\_Simulation.


 master


Aplicaciones-moviles-para-contact-tracing-usando-bluetooth / Testing caja blanca / ASI\_Simulation /

Go to file


Add file

...

 VictorJEM Carga de app replica ASI parte 4


a0b4279 1 hour ago  History

..

 .idea


Carga de app replica ASI parte 3

1 hour ago

 app


Carga de app replica ASI parte 4

1 hour ago

 dp3t-sdk


Carga de app replica ASI parte 3

1 hour ago

 gradle/wrapper


Carga de app replica ASI parte 3

1 hour ago

 .gitignore


Carga de app replica ASI parte 3

1 hour ago

 build.gradle


Carga de app replica ASI parte 3

1 hour ago

 gradle.properties


Carga de app replica ASI parte 3

1 hour ago

 gradlew


Carga de app replica ASI parte 3

1 hour ago

 gradlew.bat

Carga de app replica ASI parte 3

1 hour ago

 settings.gradle

Carga de app replica ASI parte 3

1 hour ago

Luego de abierto el proyecto, es importante realizar una modificación en la clase MainApplication.java que se encuentra donde indica la imagen de abajo.

master	Aplicaciones-moviles-para-contact-tracing-usando-bluetooth / Testing caja blanca / ASI_Simulation / app / src /		Go to file	Add file	...
		main / java / com / espol / proyecto / asi_simulation /			
VictorJEM Carga de app replica ASI parte 4			a0b4279	1 hour ago	History
..					
contacts	Carga de app replica ASI parte 4	1 hour ago			
debug	Carga de app replica ASI parte 4	1 hour ago			
extension/function	Carga de app replica ASI parte 4	1 hour ago			
inform	Carga de app replica ASI parte 4	1 hour ago			
main	Carga de app replica ASI parte 4	1 hour ago			
onboarding	Carga de app replica ASI parte 4	1 hour ago			
reports	Carga de app replica ASI parte 4	1 hour ago			
storage	Carga de app replica ASI parte 4	1 hour ago			
triage	Carga de app replica ASI parte 4	1 hour ago			
util	Carga de app replica ASI parte 4	1 hour ago			
viewmodel	Carga de app replica ASI parte 4	1 hour ago			
whattodo	Carga de app replica ASI parte 4	1 hour ago			
BaseActivity.kt	Carga de app replica ASI parte 4	1 hour ago			
MainActivity.java	Carga de app replica ASI parte 4	1 hour ago			
MainApplication.java	Carga de app replica ASI parte 4	1 hour ago			

La modificación consiste en configurar la dirección IP que actualmente utiliza la PC donde se está replicando este proyecto. Una vez identificada dicha dirección, reemplazarla en las ubicaciones que se observa en los recuadros rojos de la imagen de abajo.

```
public class MainApplication extends Application {  
  
    @Override  
    public void onCreate() {  
        super.onCreate();  
        if (ProcessUtil.isMainProcess(context: this)) {  
            registerReceiver(contactUpdateReceiver, DP3T.getUpdateIntentFilter());  
  
            PublicKey publicKey = SignatureUtil.getPublicKeyFromBase64Throw( publicKeyBase64: "LS0tLS1CRUdJTiBQVU3MSUMgS0VZLS0tLS0NCk1Ga3dFd1lIS29aSXpqMENBUVlJS29  
            DP3T.init( context: this, new ApplicationInfo( applid: "org.dpppt.demo", reportBaseUrl: "http://192.168.1.5:8085", bucketBaseUrl: "http://192.168.1.5:8085",  
            /*...*/  
        }  
    }  
}
```

Con todo lo anterior, la aplicación queda lista para instalarla en los dispositivos Android (los cuales deben estar conectados en la misma red del router casero) y comprobar su funcionamiento de intercambio de información y comunicación con el servidor para reportarse infectado con el virus y notificar a los demás usuarios con que se tuvo cercanía.