

## Guía de instalación para el despliegue de Aplicación ASI réplica

### Recursos necesarios:

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



4. Spring Tool Suite (<https://n9.cl/c3zmu>)



5. pgAdmin (<https://n9.cl/akvzl>)

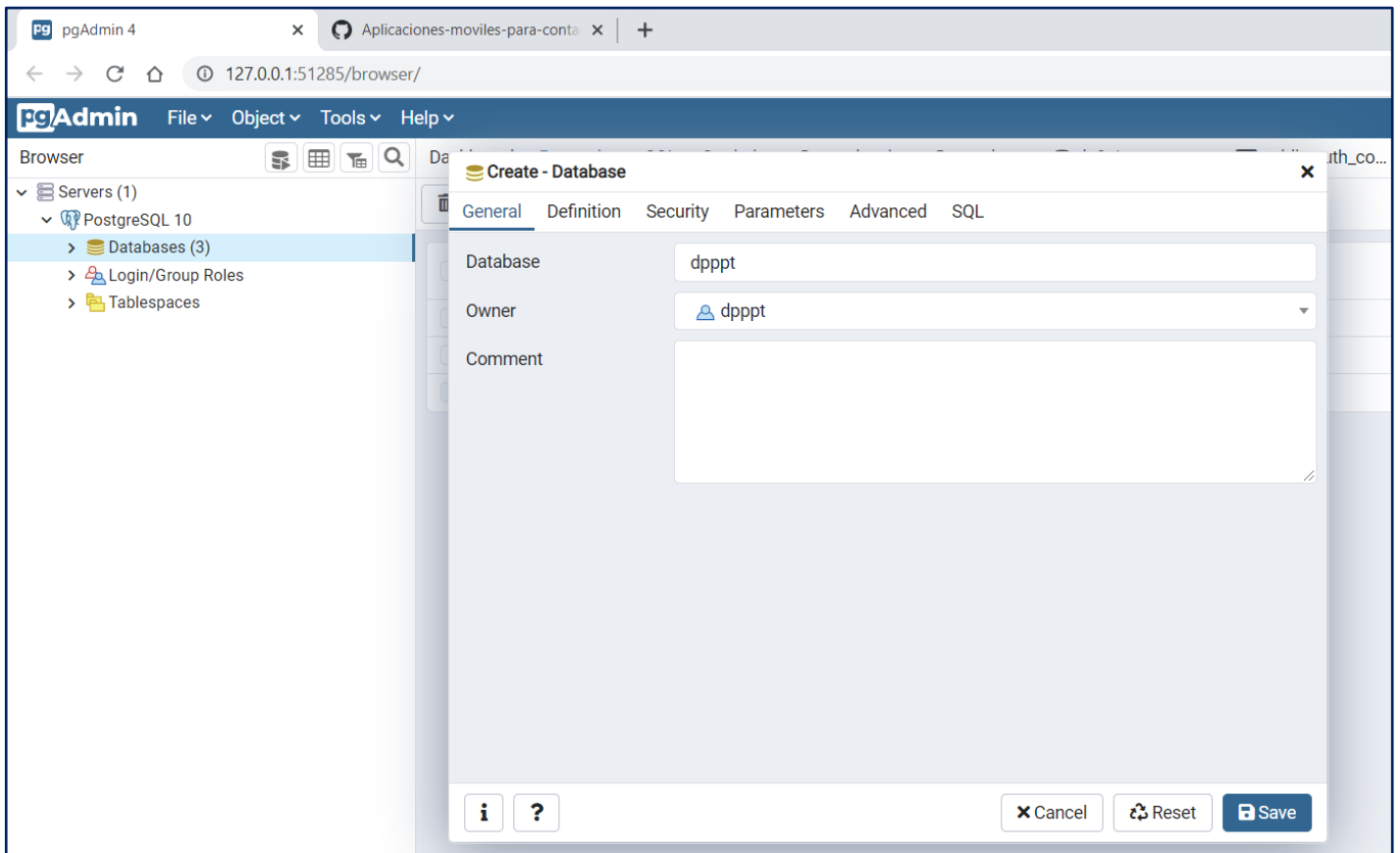


### Pasos:

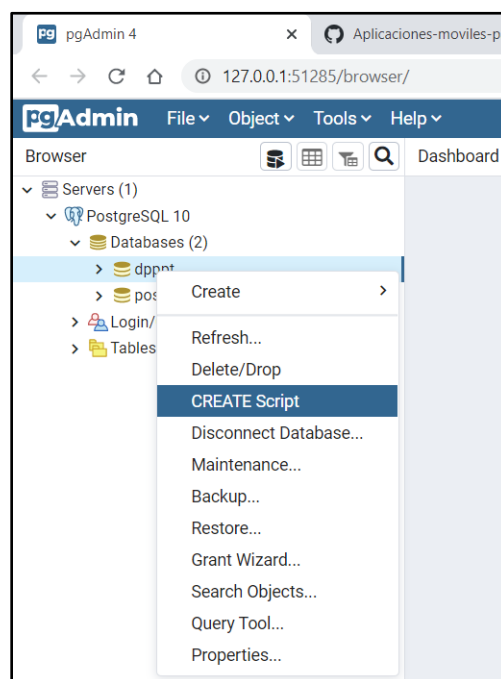
i) **Base de datos:** Se debe instalar el software de PostgreSQL en el sistema operativo (se utilizó Windows 10 en este proyecto). Crear un usuario llamado “dpppt” y contraseña “dpppt” (<https://n9.cl/e2ns0>). Por otro lado, se debe descargar el archivo **dpppt.sql** que se encuentra en el repositorio.

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	c3249d0	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	

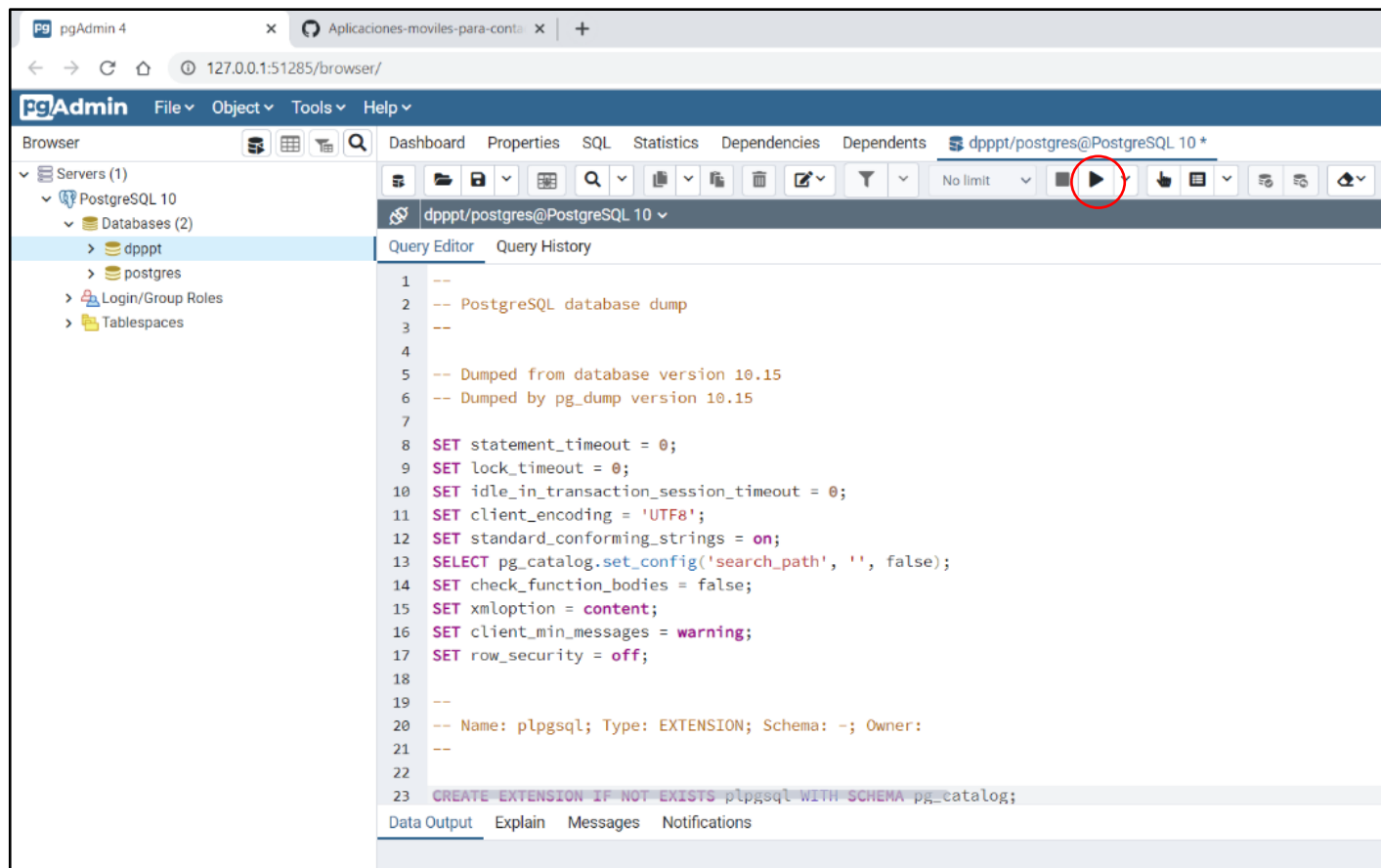
Usar pgAdmin para crear una base de datos con nombre “dpppt” y seleccionar de propietario al usuario dpppt creado en el apartado anterior.



Luego, click derecho sobre la base de datos creada y seleccionar la opción CREATE Script, lo que abrirá una sesión de Query Editor.

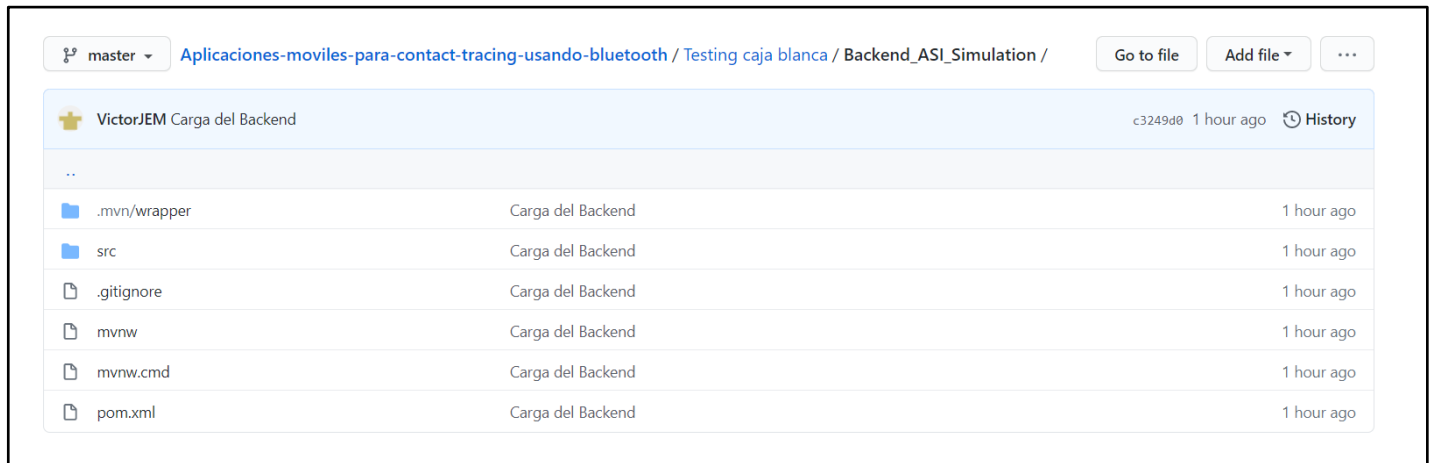


Copiar todo el contenido del archivo **dpppt.sql** descargado del repositorio y pegarlo en el Query Editor (sobrescribiendo al código SQL que aparece por defecto) como se muestra en la imagen. Por último, presionar el botón **Execute** (círculo rojo en la imagen). Con esto la base de datos queda lista y se crean las tablas e insertan los registros usados en pruebas anteriores.

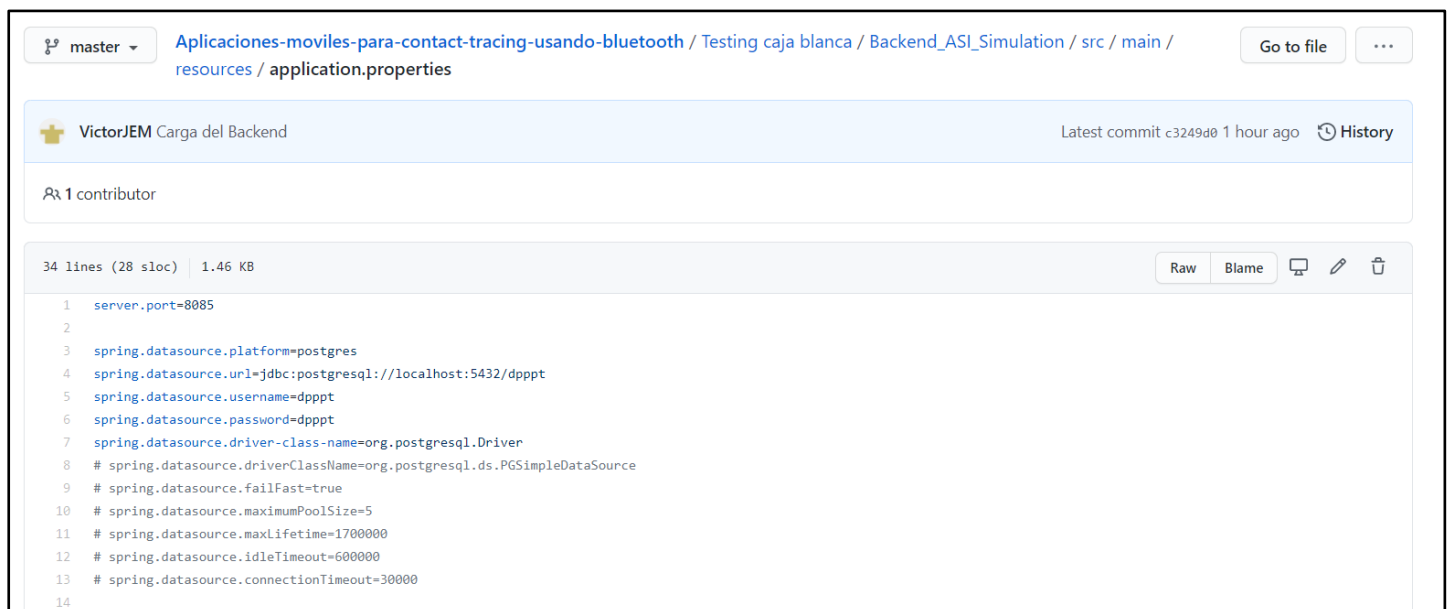


ii) **Servidor Local:** Teniendo instalado el programa Spring Tool Suite, se debe descargar la carpeta Backend\_ASI\_Simulation que contiene el código que interactúa con los usuarios y la base de datos.

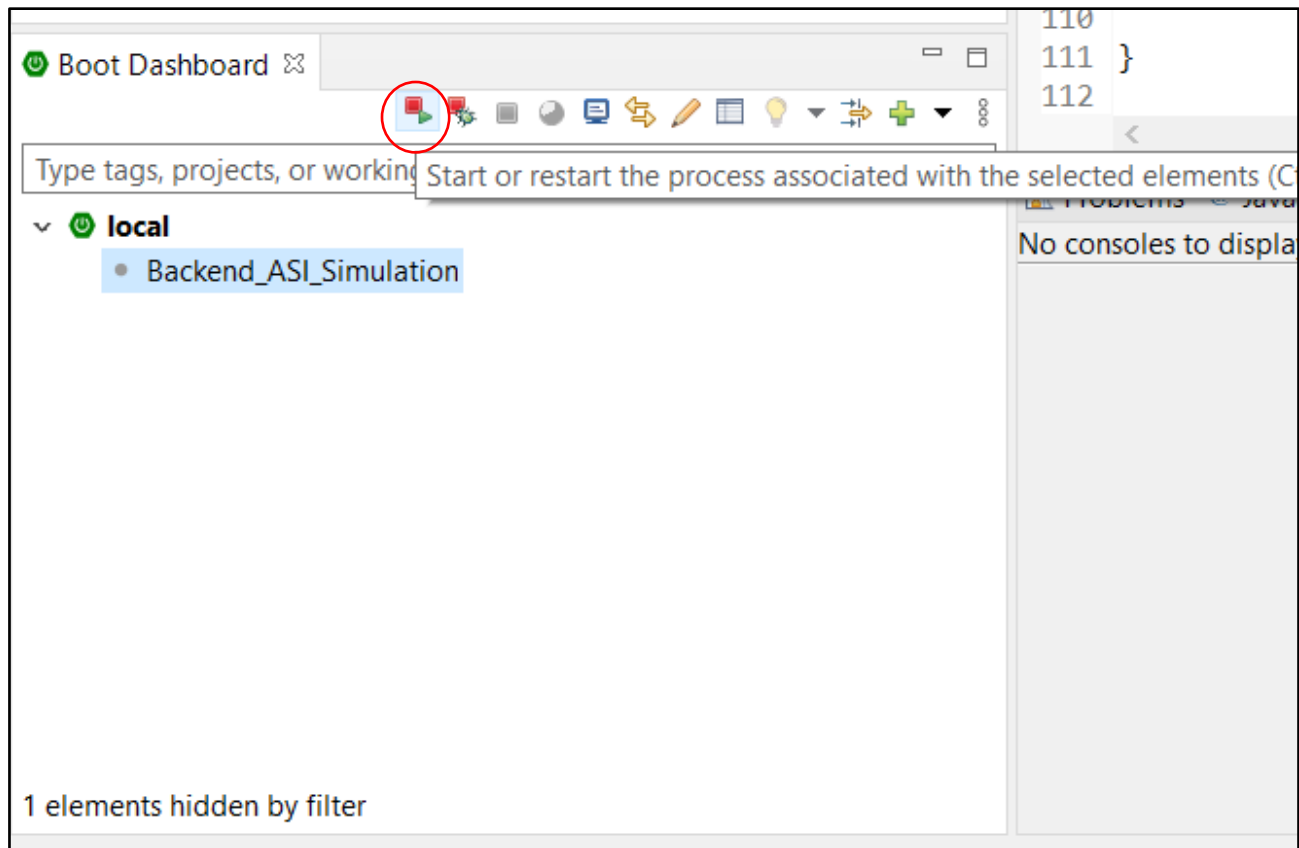
Ya en el programa se puede abrir el proyecto de manera sencilla como indica el siguiente enlace: <https://n9.cl/ayaw6>



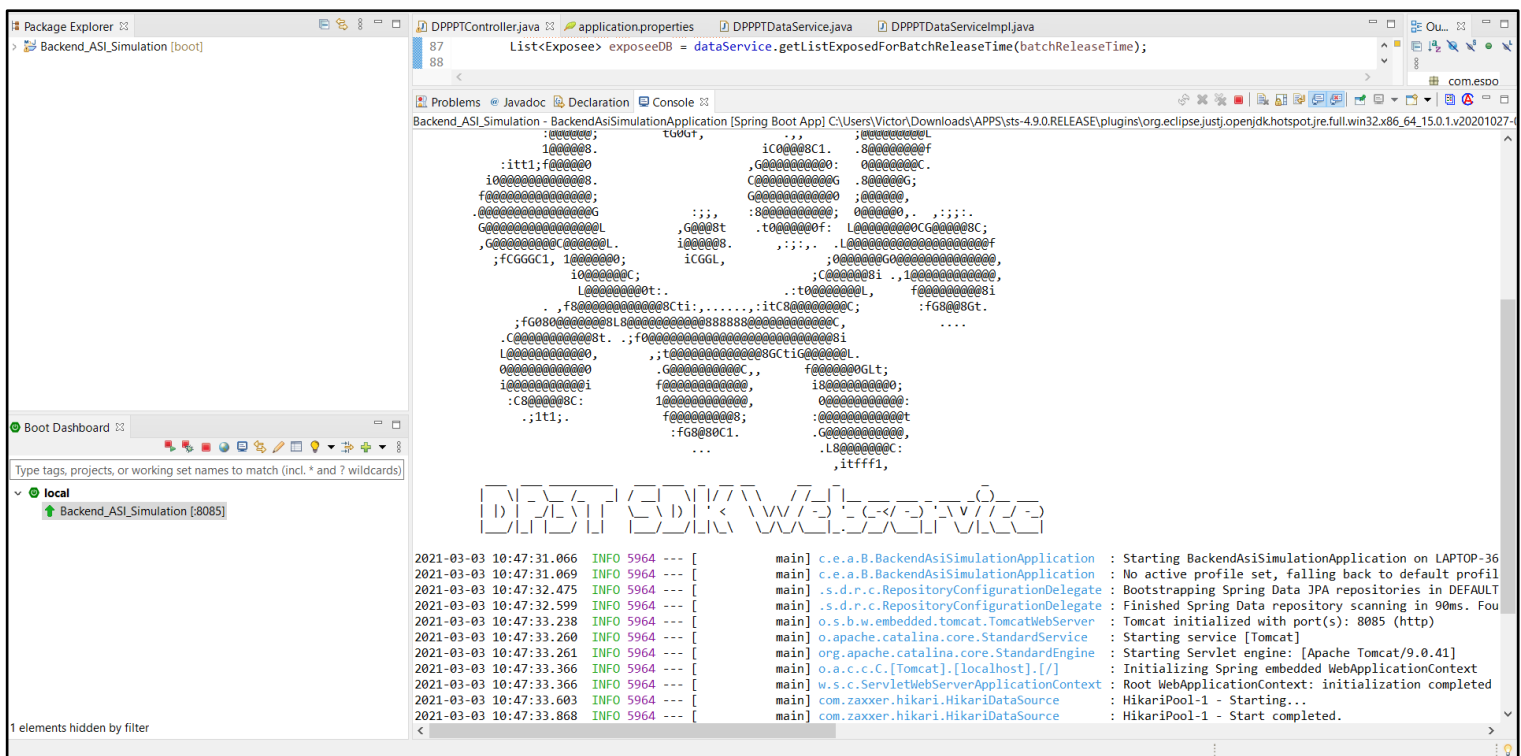
Por defecto, el servidor queda asociado con la dirección IP privada que utiliza la PC conectada a la red local del router casero. Pero es posible cambiar el puerto (server.port) que escuchará las solicitudes mediante el archivo **application.properties** como se observa en la imagen de abajo.



Una vez abierto el proyecto, ya se puede encender el servidor presionando el botón **Start** (círculo rojo) como se aprecia en la siguiente imagen:



Aquí un ejemplo de cómo se vería el servidor en ejecución:



iii) **Aplicación Réplica ASI:** 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 en Android Studio, es importante realizar una modificación en la clase MainApplication.java que se encuentra donde indica la imagen a continuación:

master ▾ [Aplicaciones-moviles-para-contact-tracing-usando-bluetooth](#) / [Testing caja blanca](#) / [ASI\\_Simulation](#) / [app](#) / [src](#) / [main](#) / [java](#) / [com](#) / [espol](#) / [proyecto](#) / [asi\\_simulation](#) / Go to file Add file ▾ ...

VictorJEM Carga de app replica ASI parte 4 a0b4279 1 hour ago [History](#)

..

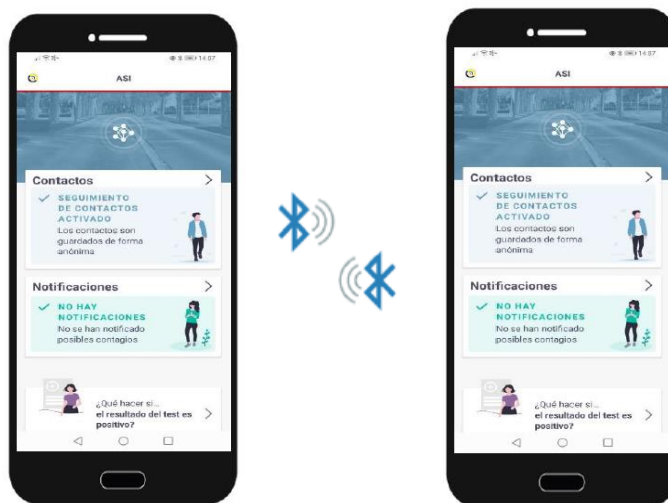
contacts	Carga de app replica ASI parte 4	1 hour ago
debug	Carga de app replica ASI parte 4	1 hour ago
extension/function	Carga de app replica ASI parte 4	1 hour ago
inform	Carga de app replica ASI parte 4	1 hour ago
main	Carga de app replica ASI parte 4	1 hour ago
onboarding	Carga de app replica ASI 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 (conectada al router casero) 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.getPublicKeyFromBase64OrThrow( publicKeyBase64: "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0NCk1Ga3dFd1lIS29aSXpqMENBUVlJS29  
            DP3T.init(context: this, new ApplicationInfo( appld: "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 a la misma red) y comprobar su funcionamiento.

**iv) Comprobación:** Cabe mencionar que los dispositivos deben estar cerca por al menos 15 minutos para que la información intercambiada quede registrada correctamente.



Pasado el tiempo, copiar el código COVID de uno de los dispositivos y registrarlo mediante pgAdmin en la tabla **auth\_codes** de la base de datos **dpppt**.



Dashboard Properties SQL Statistics Dependencies Dependents public.auth\_co... publi

dpppt/postgres@PostgreSQL 10

Query Editor Query History

```

1 INSERT INTO public.auth_codes(
2     id, value)
3     VALUES (6, 'XLU357UZK3');
```

Data Output Explain Messages Notifications

INSERT 0 1

Query returned successfully in 59 msec.

Después del registro, presionar el botón Notificar anónimamente y con esto queda registrada la información de dicho usuario en la otra tabla de la base de datos.

Dashboard Properties SQL Statistics Dependencies Dependents public.auth\_co... public.exposee... dpppt/postgres... publi

public.exposees/dpppt/postgres@PostgreSQL 10

Query Editor Query History

```

1 SELECT * FROM public.exposees
2 ORDER BY id ASC
```

Data Output Explain Messages Notifications

id [PK] integer	canton character varying (255)	key character varying (255)	key_date bigint	parroquia character varying (255)	provincia character varying (255)
1	1 GUAYAQUIL	LJMaX+P79unB7QdzJnY8A16RTW8U0anhbhlDbDeGYWM=	1609804800000	NUEVE DE OCTUBRE	GUAYAS
2	2 MANTA	C/qT9Ny2/4A+V57t03stVAW20ZFt9UgGg8aO2okdGb4=	1609804800000	ELOY ALFARO	MANABI
3	3 QUEVEDO	h1KQe9R5fJAiTkeU9dKy1A5DvrYsmOeP6tn0DrsEDHE=	1609804800000	GUAYACÁN	LOS RIOS
4	4 LA MANÁ	v3RCnd2BTV7EfelYc9t2qqU8pwPZnXG2OvZ6Dm0egQc=	1609891200000	GUASAGANDA (CAB.EN...	COTOPAXI
5	5 GUAYAQUIL	Hwb0moUVaTEmhQR1WcWzf7Klht3aCBFu9AKf9zQoWsE=	1610064000000	NUEVE DE OCTUBRE	GUAYAS
6	6 DURÁN	Ewk2n4qyzJ/mAbkLvg6HFNc2qKfcyK/90raojN8oBQk=	1612915200000	EL RECREO	GUAYAS



Para obtener la notificación de posible contagio de manera inmediata en el otro dispositivo basta con abrir y cerrar la aplicación para que haga la consulta al servidor y compare con sus contactos, y cuando encuentra una coincidencia muestra un mensaje como se observa en la siguiente imagen.

