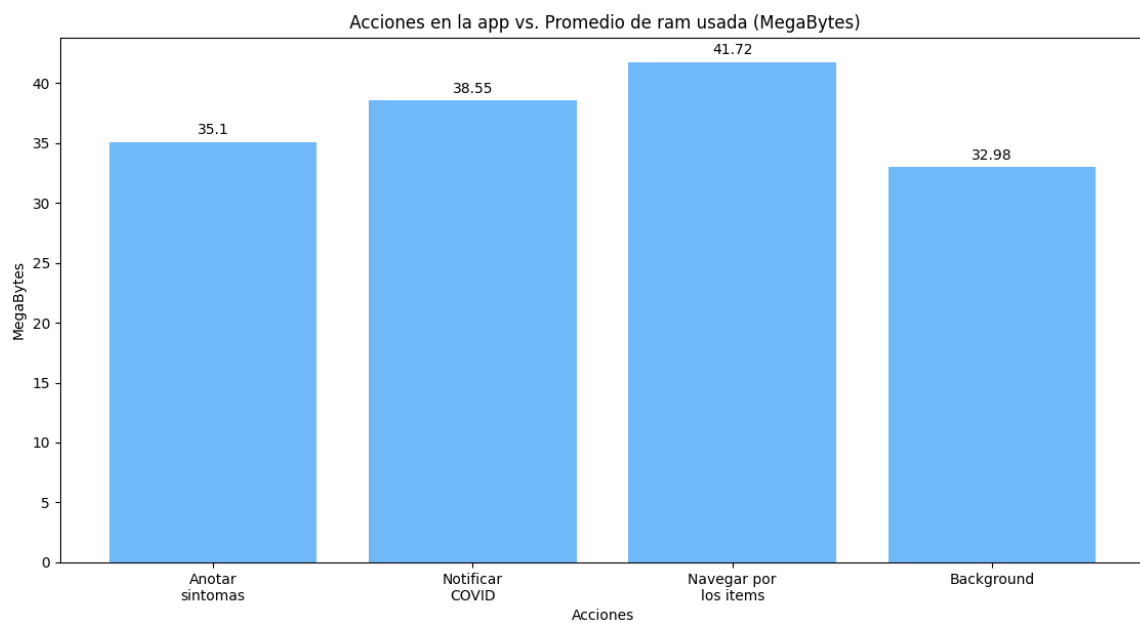
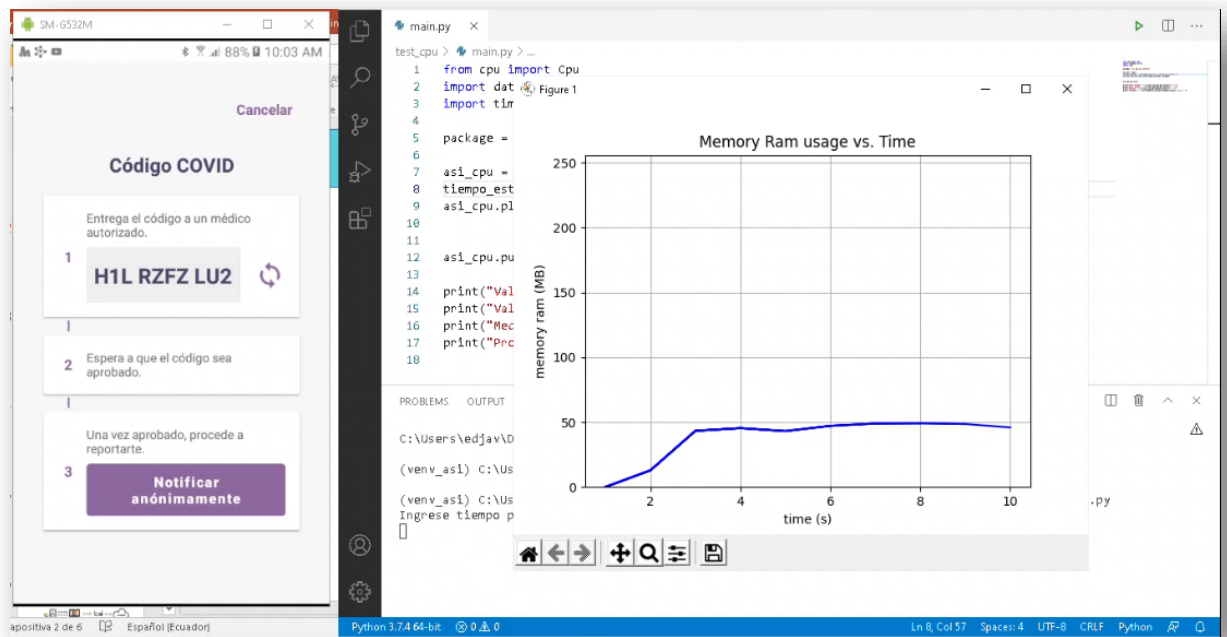


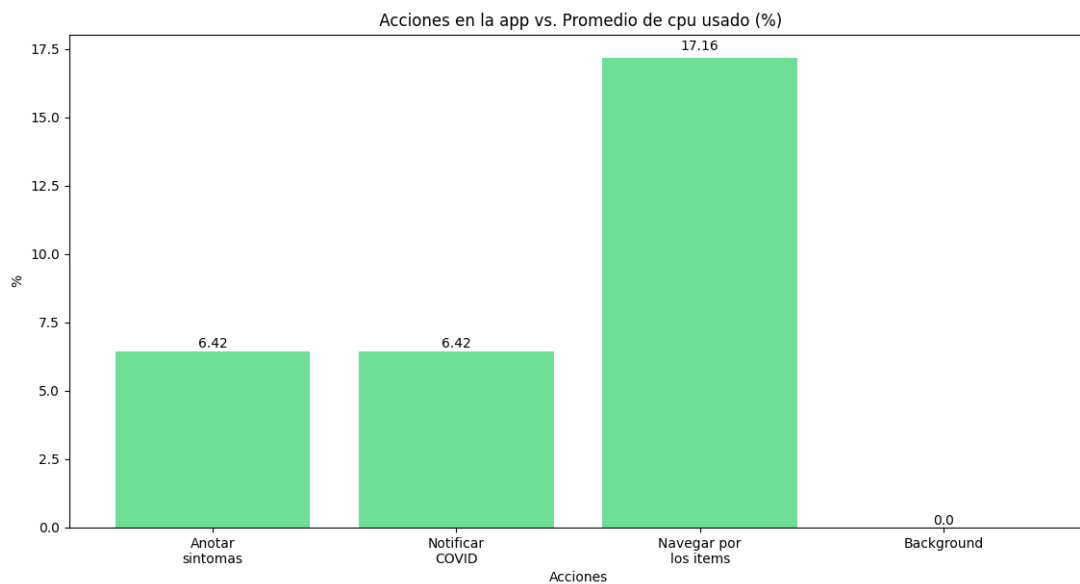
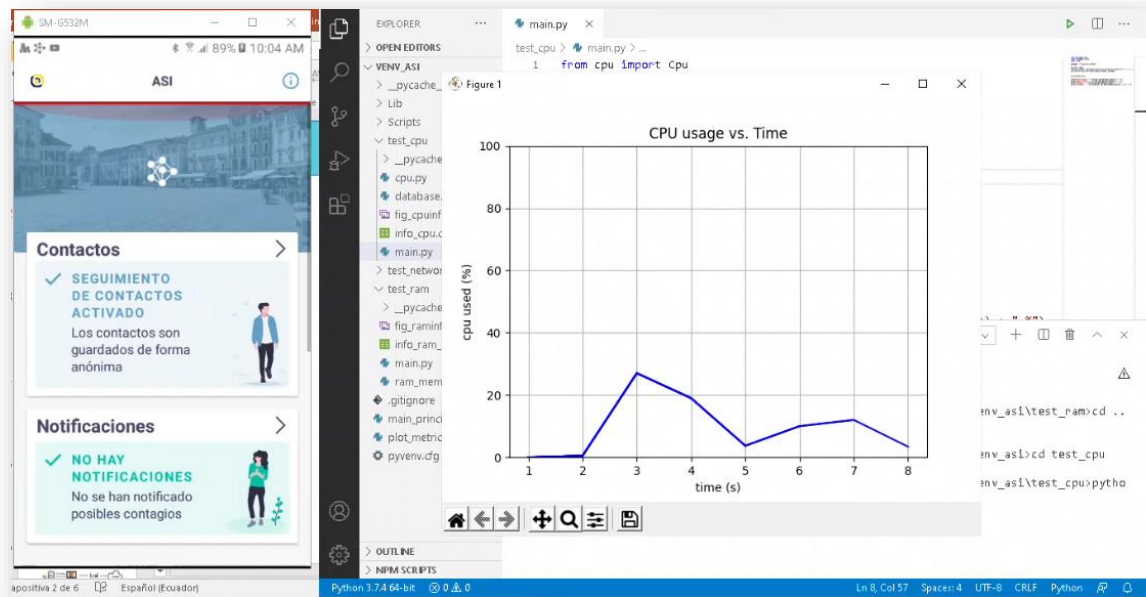
Análisis de caja negra

- Test de memoria RAM usado



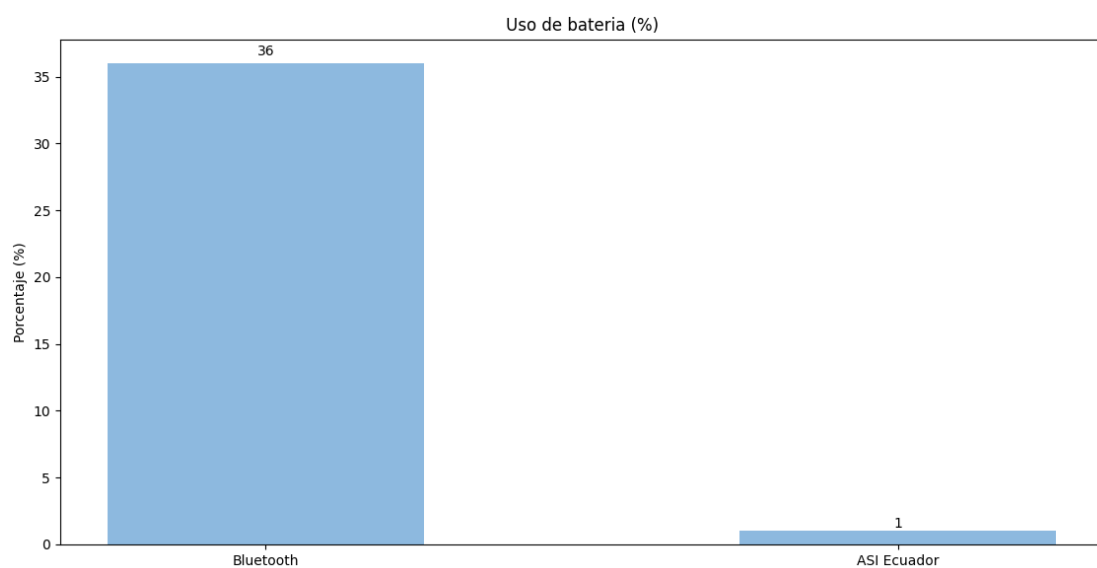
Se realizaron 20 ejecuciones del script en Python por las cuatro acciones que se observan en la figura anterior, logrando cuantificar el promedio de memoria RAM usada por cada acción.

- Test de CPU usado



Se realizaron 20 ejecuciones del script en Python por las cuatro acciones que se observan en la figura anterior, logrando cuantificar el promedio de CPU usado por cada acción.

- Recolección del uso de batería



Se recogieron 5 veces los valores de batería usado por la aplicación móvil dejando pasar un día, logando cuantificar el promedio de batería usada.

- Burp Suite Community E... 10:16 PM 🔊 📶 76% 🔒

Burp Suite Community Edition V2020.9.1 - Temporary Project

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type...	Extension	Title	Comment	TLS	IP	Cookies	Time	Listener port
1	https://142.250.64.163	GET	/generate_204		✓	204	102					✓	142.250.64.163		21:54:50.1	8000
2	https://play.googleapis.com	POST	/logbatch		✓	200	502	text				✓	172.171.3.138		21:54:50.1	8000
3	https://play.googleapis.com	POST	/logbatch		✓	200	539	text				✓	172.171.3.138		21:54:52.1	8000
4	https://service.gcm-mode.net	GET	/gcm/model/deviceval/glob?at=		✓	825	825	JSON				✓	67.202.16.36	SCOUTER=x1q.	21:54:52.1	8000
5	https://service.gcm-mode.net	GET	/gcm/model/dl/packages/device...		✓	200	5662	JSON				✓	67.202.16.36	SCOUTER=x52.	21:54:56.1	8000
6	https://connect.raing.covidanalytics.ai	POST	/v1/generate/inset		✓	200	1845	JSON				✓	190.152.52.235		21:55:03.1	8000
7	https://connectivitycheck.gstatic.com	GET	/generate_204		✓	204	280					✓	142.250.64.163		21:55:10.1	8000
9	https://googleads.g.doubleclick.net	GET	/madm/gma?admobmodel=SM-G532...		✓	200	63991	HTML				✓	142.250.64.194		21:55:30.1	8000
10	https://firebaseerrorconfig.googleapis.com	POST	/v1/projects/BG642449822/nam...		✓	503	563	JSON				✓	172.171.15.202		21:55:30.1	8000
11	https://googleads.g.doubleclick.net	GET	/actconfig/pubsetting/wap_name...		✓	200	1118	JSON				✓	142.250.64.194		21:55:31.1	8000
12	https://www.googleadservices.com	GET	/activeview/js/current/fidaz.js?ca...		✓	200	78206	script		js		✓	172.171.2.194		21:55:33.1	8000
13	https://connectivitycheck.gstatic.com	GET	/generate_204		✓	204	280					✓	142.250.64.163		21:55:40.1	8000
14	https://connectivitycheck.gstatic.com	GET	/generate_204		✓	204	280					✓	142.250.64.163		21:55:44.1	8000
15	https://connect.raing.covidanalytics.ai	POST	/v1/generate/exposed		✓	507	527	text				✓	190.152.52.235		21:55:46.1	8000
18	https://connect.raing.covidanalytics.ai	GET	/v1/config/appversion/android-0...		✓	200	1180	JSON				✓	190.152.52.235		21:56:10.1	8000

The screenshot shows the 'Request' tab in a web browser's developer tools. The request is a POST to /v1/gaen/onset. The headers are: accept: */*, User-Agent: ec.gob.asi.android;0.0.12-pilot;1598659886251;Android;23, Content-Type: application/json, Content-Length: 43, Host: contacttracing.covidanalytics.ai, Connection: close, and Accept-Encoding: gzip, deflate. The body is a JSON object: {"authorizationCode": "HGF594794U", "fake": 0}.

The screenshot shows a web browser window with the address bar displaying the URL: `https://contacttracing.covidanalytics.ai/v1/green/exposed`. The browser's developer tools are open, showing the 'Network' tab with a single request. The 'Response' pane displays a large JSON object. The JSON structure includes fields like `Authorization: Bearer`, `User-Agent`, `Content-Type`, `Content-Length`, `Host`, `Connection`, and `Content-Encoding`. The main data is a `keyframes` array, which contains a single object with a `keyframe` array. This array contains several objects, each with a `keyframe` array of its own, representing a sequence of events or data points over time.

- Test de red usado por la aplicación móvil

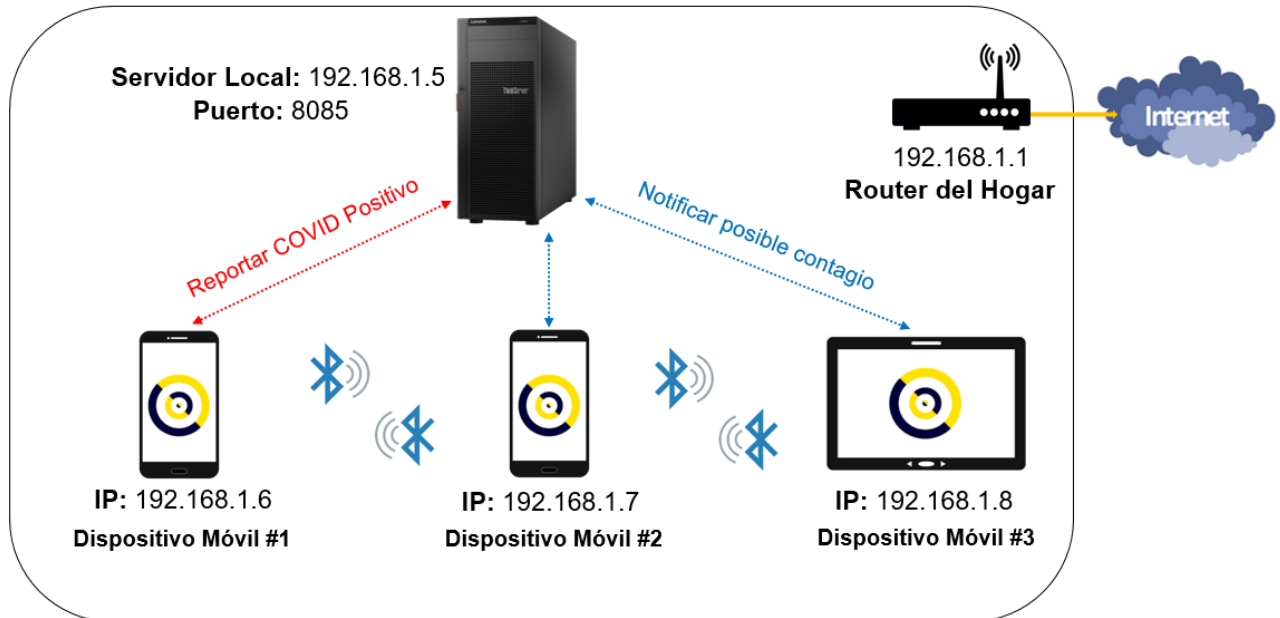


Datos de red de subida y bajada usados por la aplicación móvil al ejecutar por primera vez y cuando se valida el código COVID.

Análisis de caja blanca

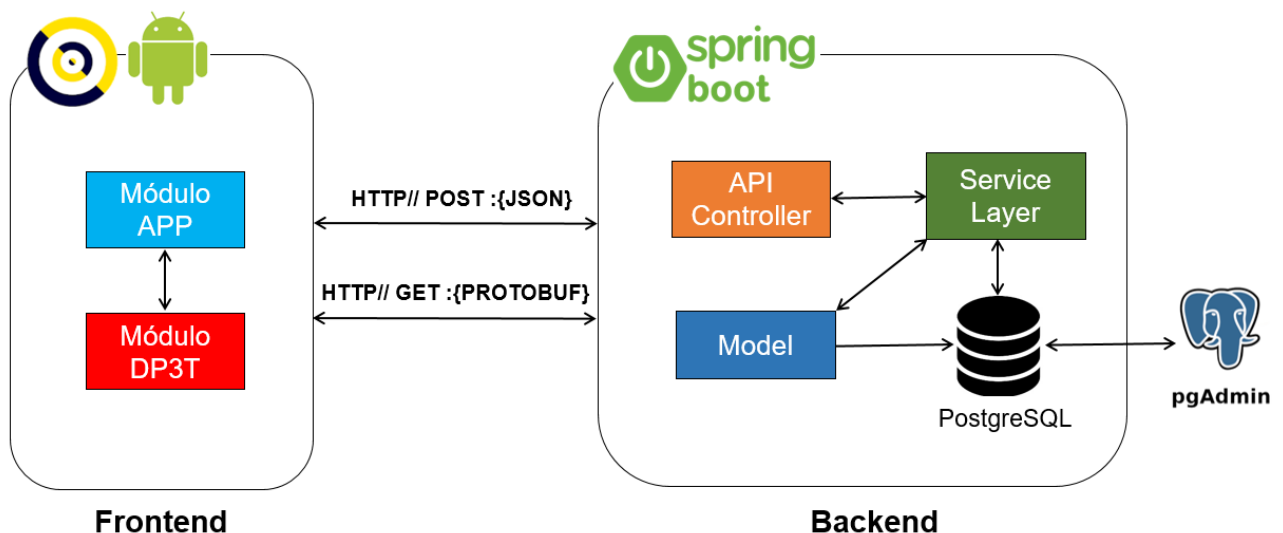
Arquitectura de red

En la imagen se muestran los componentes usados para el entorno simulado. Las pruebas de caja blanca consistieron en estudiar el proceso de intercambio de llaves mediante Bluetooth y la comunicación con el servidor local para reportarnos como COVID positivo y también para notificar de un posible contagio a los contactos.

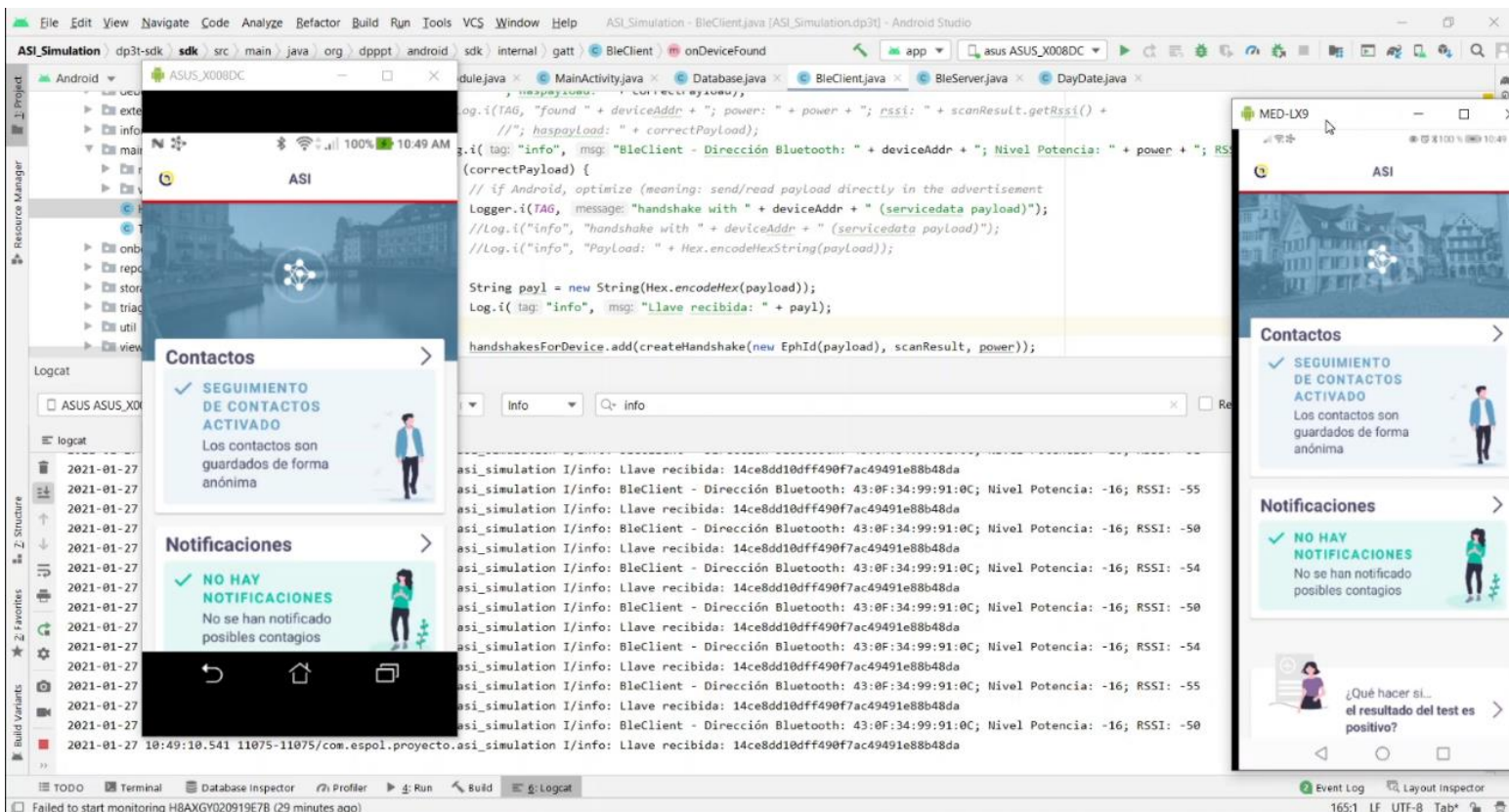
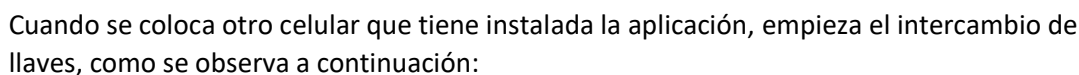


Arquitectura de servicios

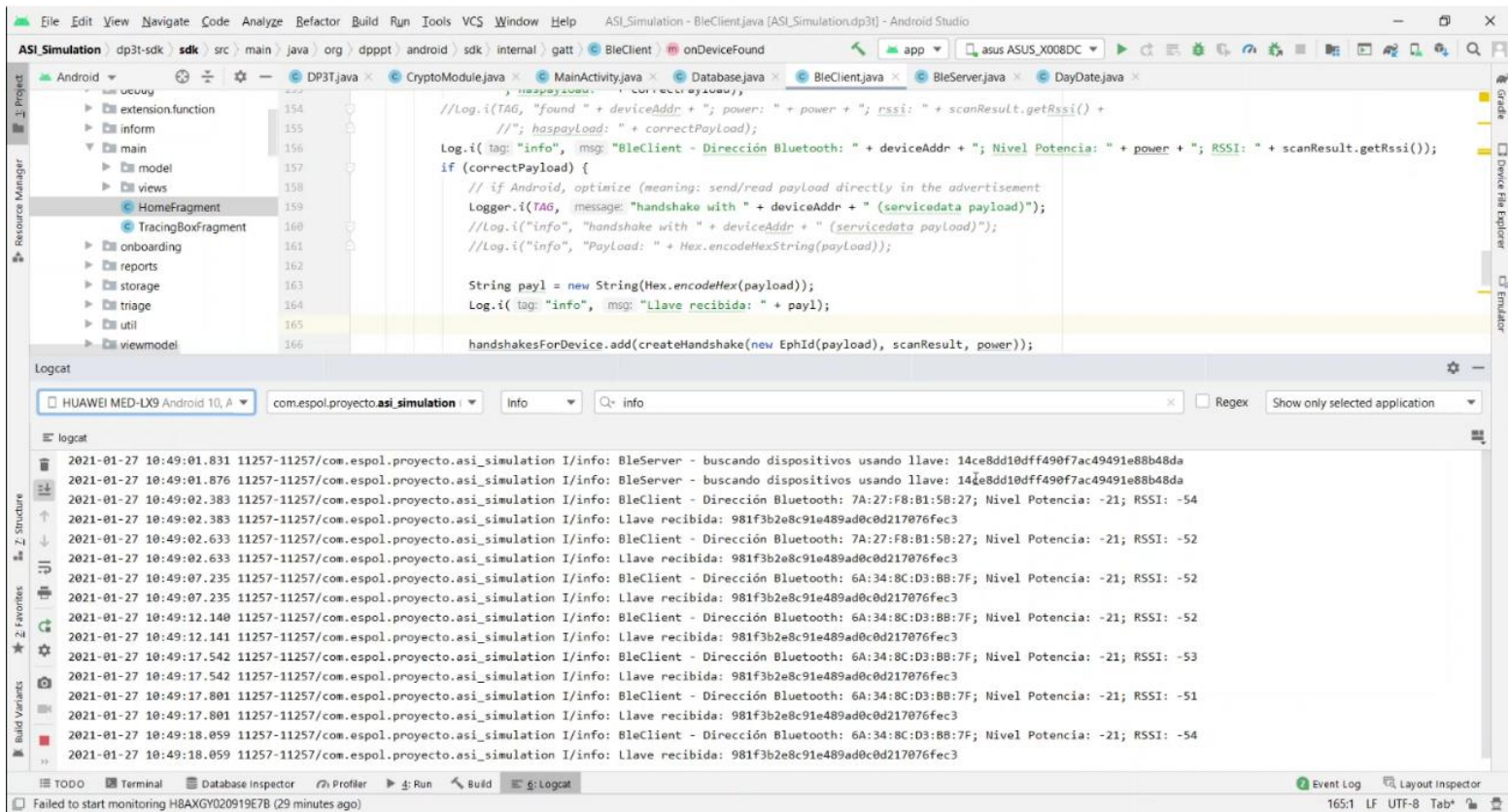
El frontend cuenta con dos módulos, APP que tiene el código para la apariencia y DP3T que tiene las funcionalidades para el rastreo de contactos y comunicaciones con la API. El backend está implementado bajo el framework Spring Boot con el patrón MVC y se comunica con su base de datos PostgreSQL, la cual es administrada mediante pgAdmin.



Mediante la herramienta Logcat de Android Studio se imprimen las llaves que utiliza el primer celular para intercambiar con otros dispositivos.

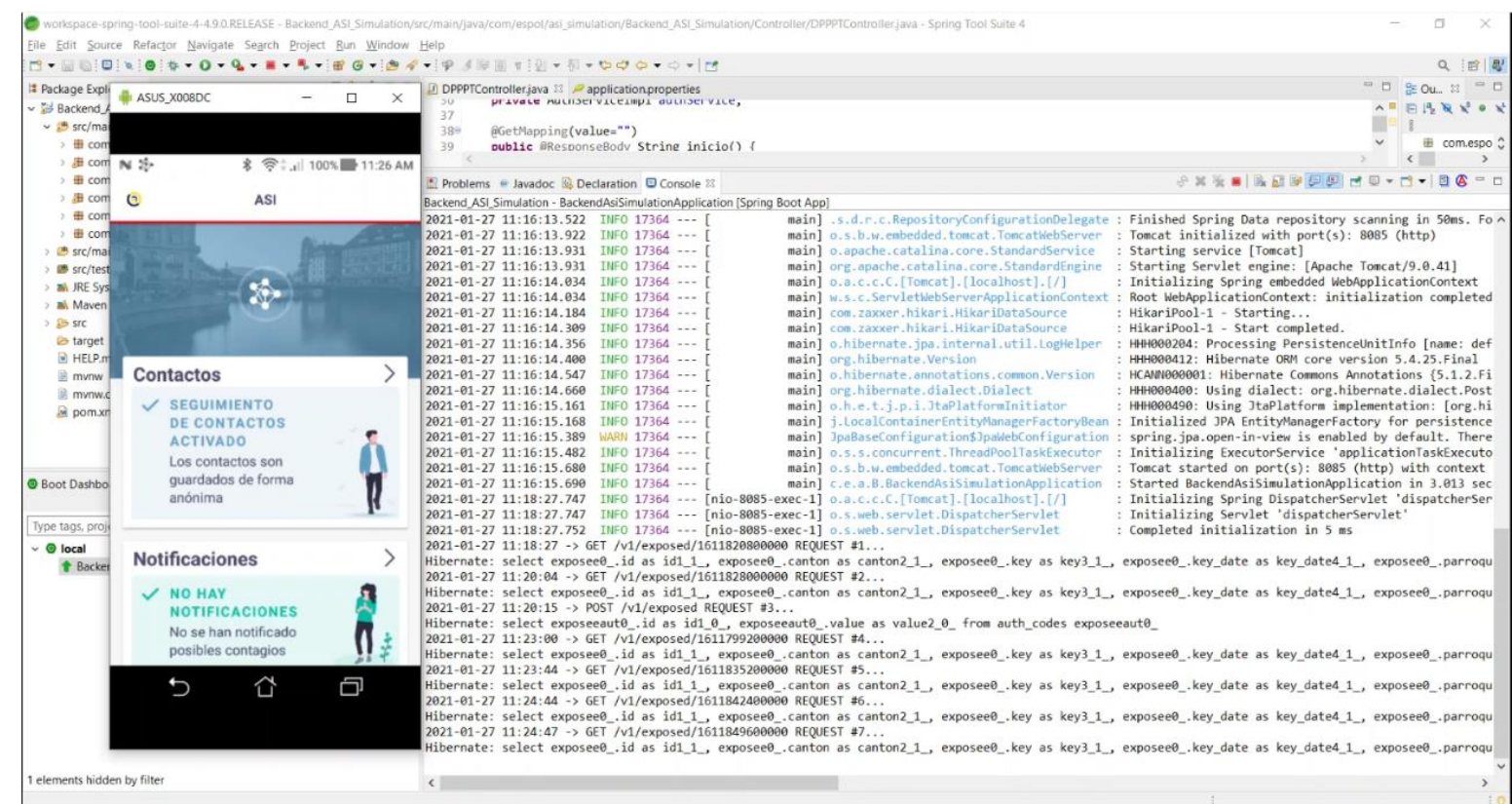


Aquí se muestra, lo que recibió el segundo dispositivo móvil:

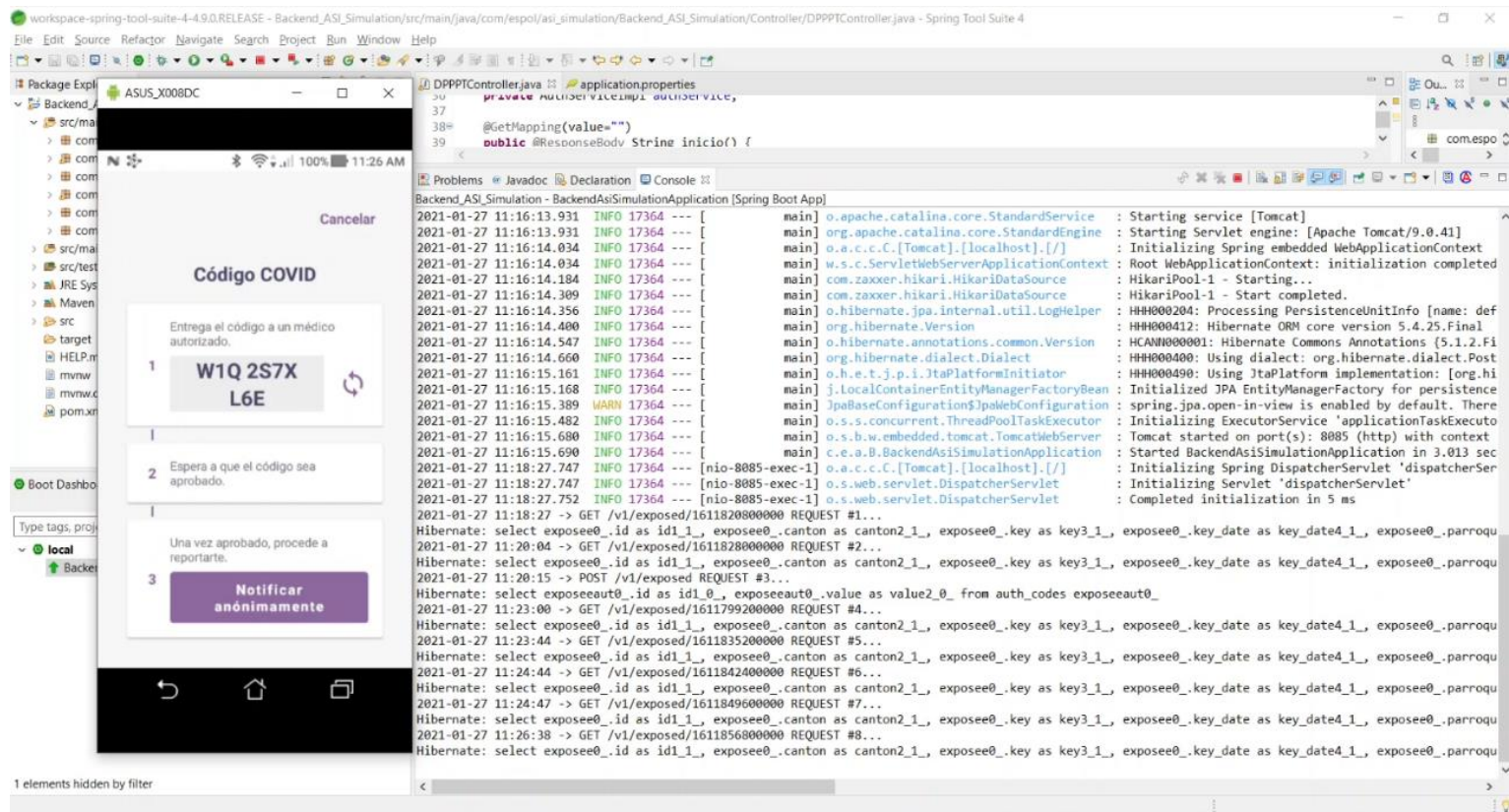


Resultados: Comunicación con el servidor

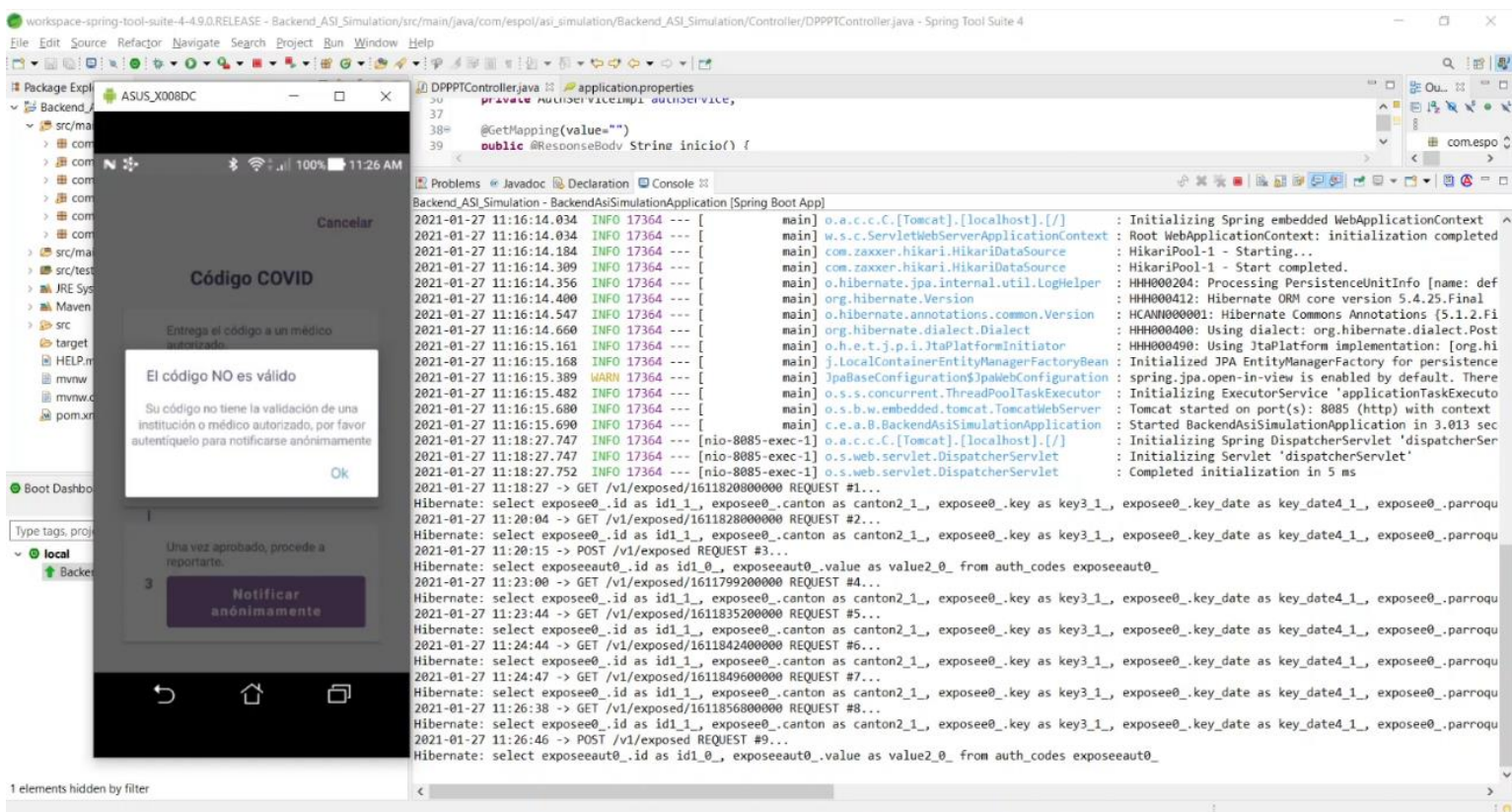
La siguiente captura muestra el primer dispositivo junto con el programa Spring Tool Suite, donde se desplegó el servidor local.



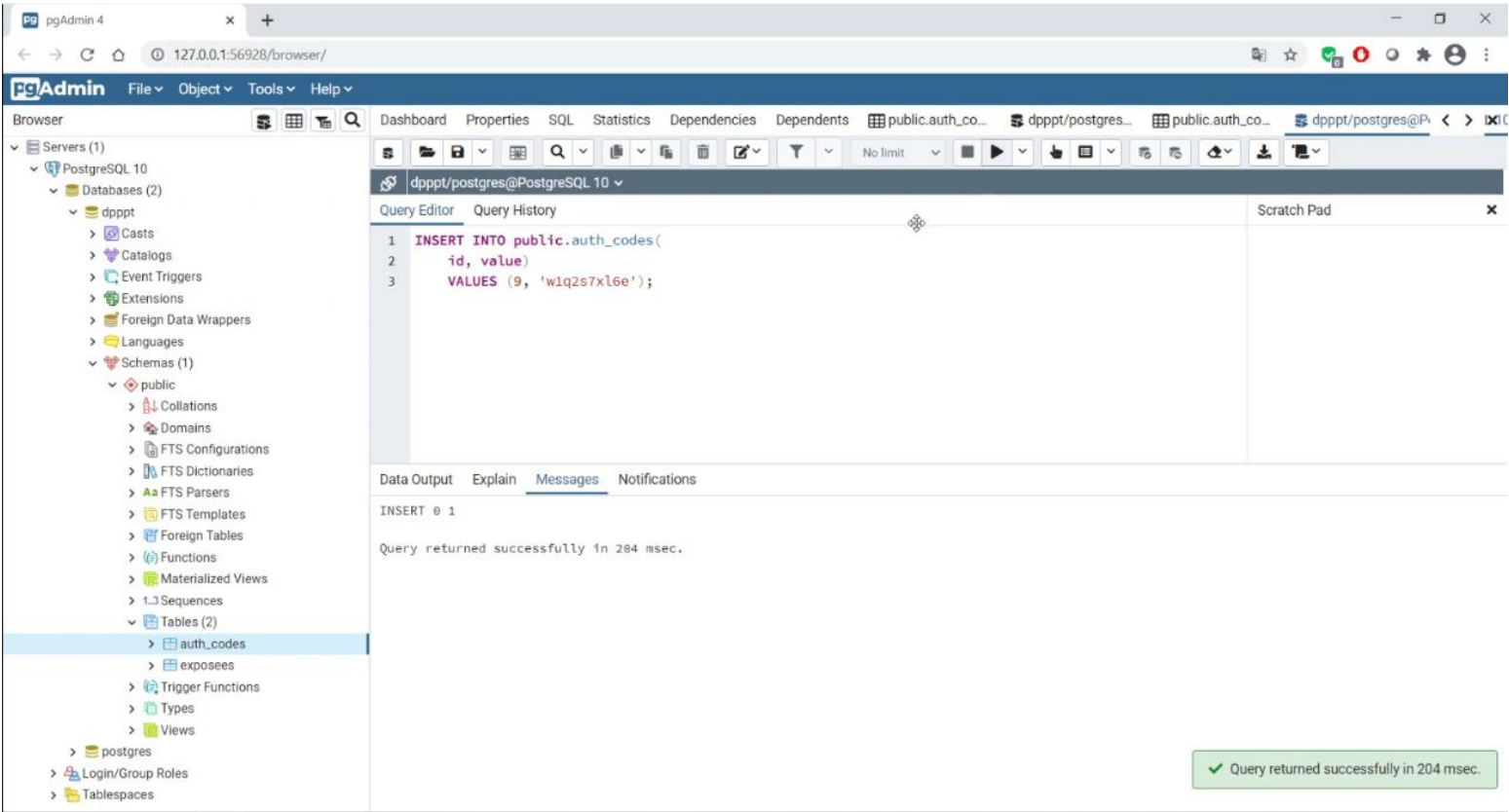
Aquí se usa la opción para reportarse como COVID positivo, donde también se destaca el código **W1Q2S7XL6E**, el cual debe ser ingresado en la base de datos.



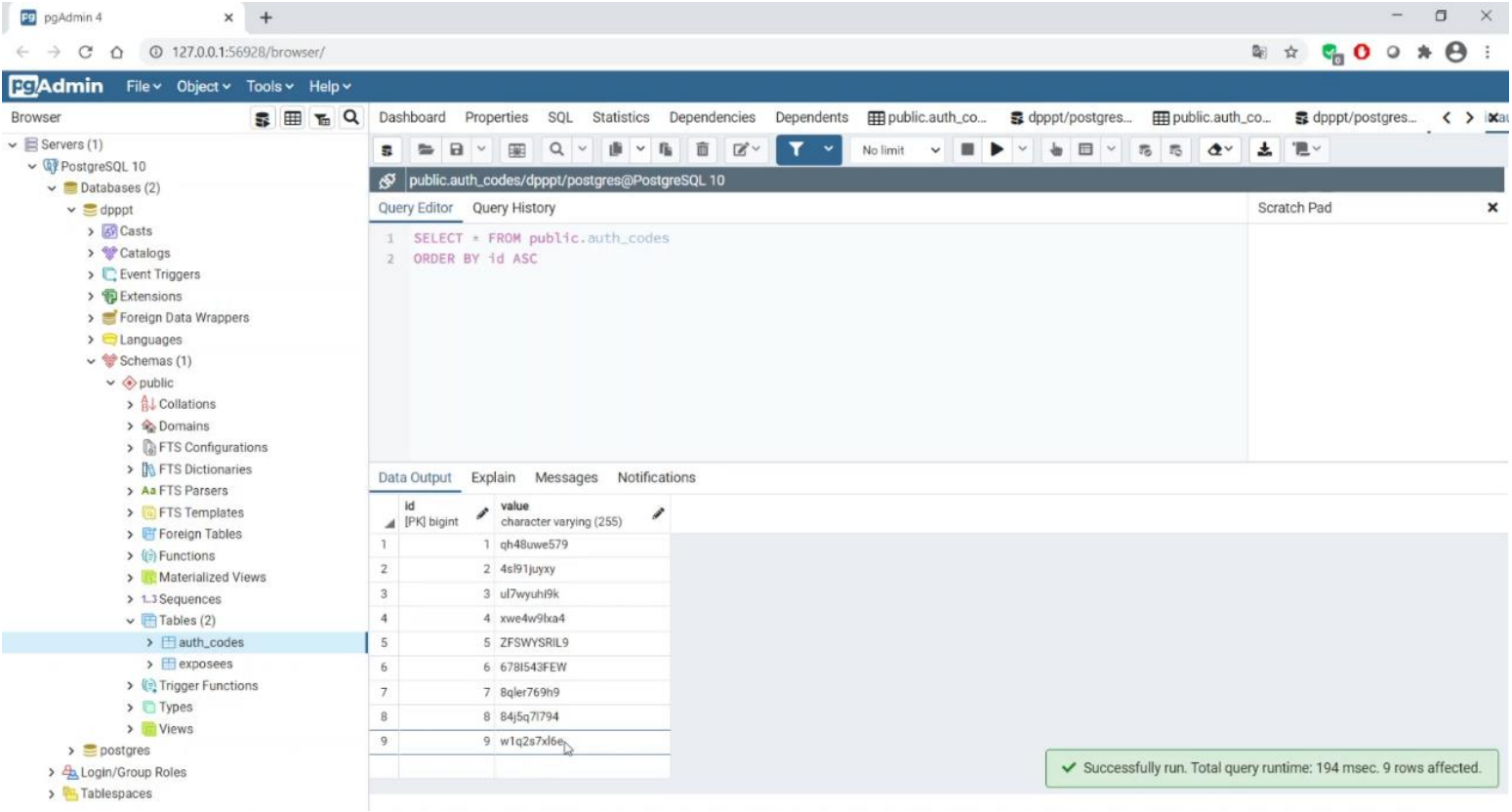
Al presionar el botón **Notificar anónimamente**, el servidor responde que el código NO es válido, debido a que aún no se ha registrado el código COVID que se mencionó anteriormente.



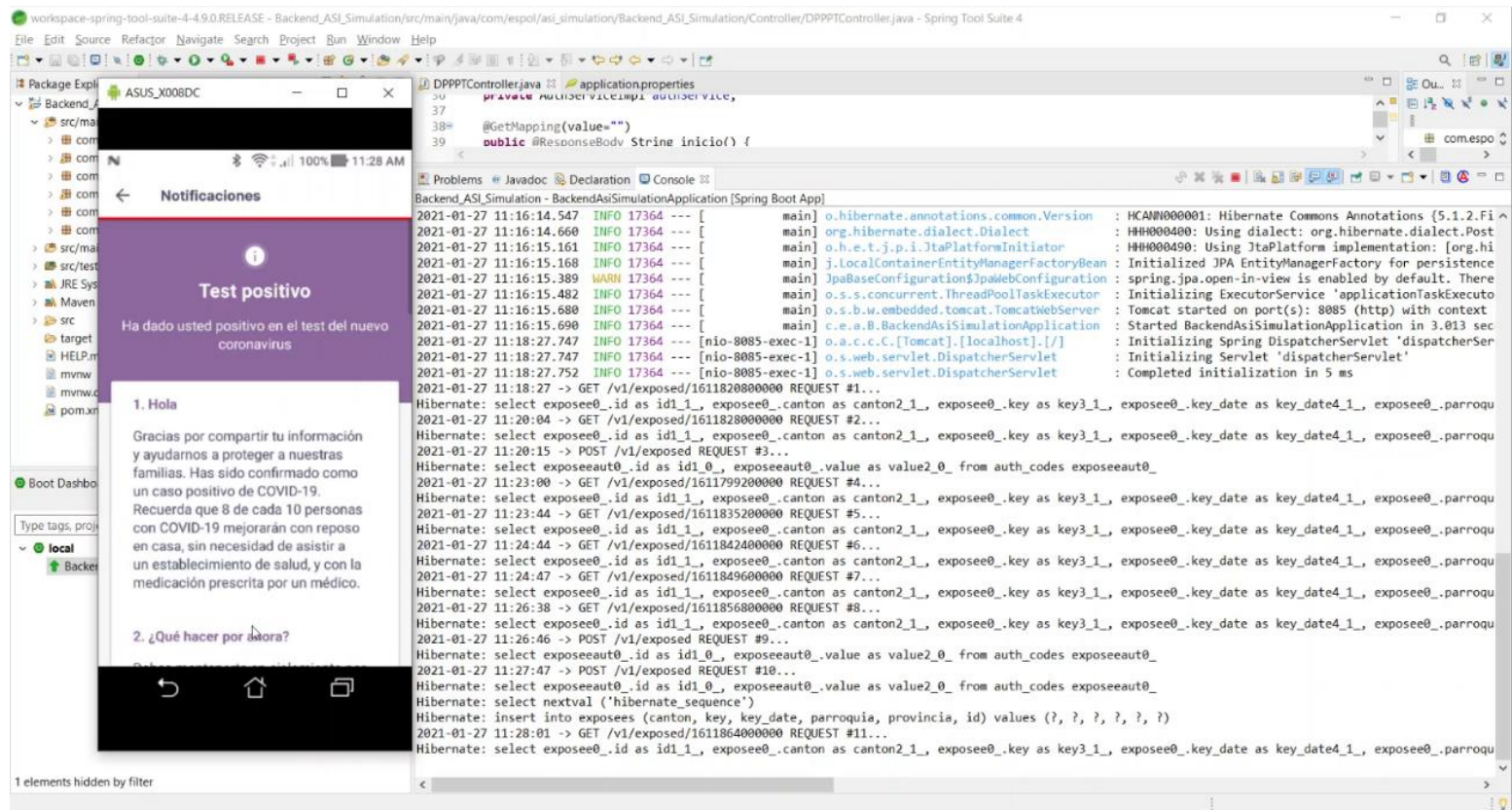
Para el registro del código se utiliza la interfaz web pgAdmin, y en la siguiente imagen se observa el query ingresado en la tabla auth_codes.



Se consultan todos los valores de la tabla y en la novena posición se visualiza el código correctamente ingresado, con lo cual el usuario ya podrá reportarse.



Luego se repite el proceso y esta vez sí responde favorablemente el servidor y aparece un mensaje de test positivo.



Finalmente, se observa que en el segundo dispositivo llega la notificación de posible contagio.

