**Documento de Arquitectura de Software**

***Sistema de monitoreo de***

***zona de riesgo de contagio por Covid-19***

***Versión 1.0***

***Fecha: 15/07/2020***

**ÍNDICE**

[**Propósito de la solución**](#_heading=h.gjdgxs) **3**

[**Definiciones y convenciones**](#_heading=h.30j0zll) **3**

[**Antecedentes**](#_heading=h.1fob9te) **4**

[**Arquitectura general de la solución**](#_heading=h.1t3h5sf) **4**

[Arquitectura de la solución](#_heading=h.2s8eyo1) 4

[Tecnologías utilizadas](#_heading=h.17dp8vu) 4

[**Vistas de la solución**](#_heading=h.1u6ztwuhp333) **6**

[Vista modular](#_heading=h.3rdcrjn) 6

[Vista de despliegue](#_heading=h.35nkun2) 7

## **Propósito de la solución**

Brindar una herramienta que informe en tiempo real acerca de las zonas de riesgo, donde además de la visualización de información, se tendrá en un seguimiento del estado de salud de cada usuario de la aplicación, además de recomendaciones e información actualizada según su salud actual.

## **Definiciones y convenciones**

|  |
| --- |
| **Mapa de calor:** Un mapa de calor es una técnica de visualización de datos que muestra la magnitud de un fenómeno mediante un código de colores. |
| **Malestar:** Sensación física de indisposición, que puede tener causas que no se deben a una enfermedad subyacente. |
| **COVID-19:** Es una enfermedad infecciosa causada por un ‎coronavirus recientemente descubierto**.** Afecta a los infectados de distintas maneras. La mayoría de las personas que se contagian presentan síntomas de intensidad leve o moderada**.** |
| **Geolocalización:** Consiste en **obtener la ubicación geográfica** de un objeto como puede ser un teléfono móvil, un coche o un una calle. Para ello se puede utilizar diferentes métodos como por ejemplo comprobar el código postal de una carta, la dirección IP de un equipo o el sistema GPS de nuestro teléfono móvil. |
| **APK:** Archivos **ejecutables** para Android. Beben del formato .jar de Java, y permiten instalar componentes empaquetados en el sistema |
| **HTTP:** El Protocolo de transferencia de hipertexto es el **protocolo de comunicación** que permite las transferencias de información en la World Wide Web. |

## **Antecedentes**

En la actualidad nos enfrentamos a una situación nunca antes vivida por nuestra generación. La pandemia de la enfermedad Covid-19, misma que es altamente contagiosa, se expande incluso sin tener contacto directo con personas afectadas por este mal.

La información respecto a esta enfermedad es poca, debido a su reciente brote, es por ello que las medidas preventivas son necesarias para evitar su expansión, evitando exponernos a lugares concurridos así como mantener una limpieza continua.

Por esto, es importante contar con una herramienta que brinde información en tiempo real sobre el desarrollo de dicha enfermedad en zonas aledañas a nuestros hogares o a sitios que concurrimos regularmente para evitar futuros contagios hacia nosotros o a los nuestros.

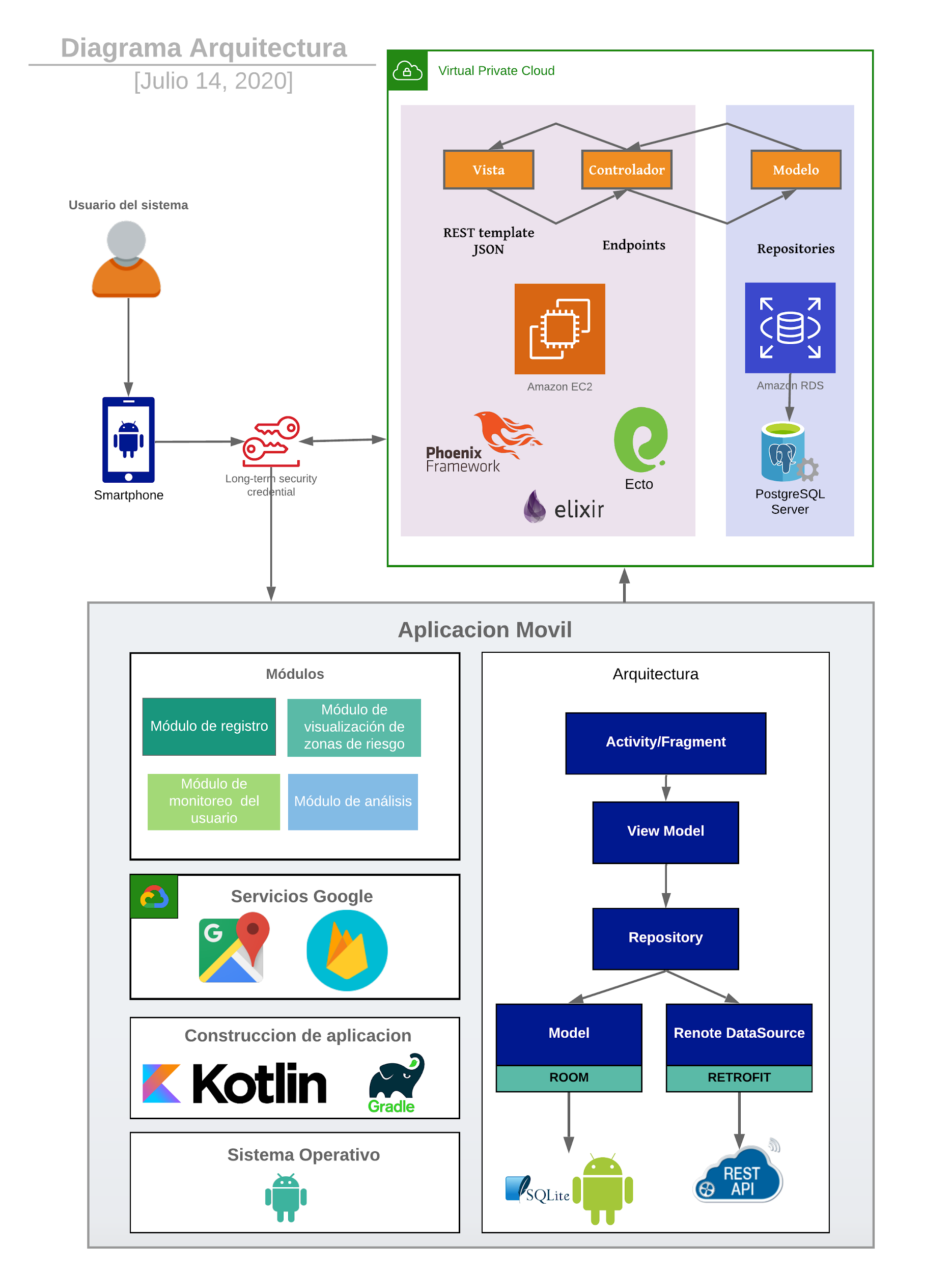
## **Arquitectura general de la solución**

### Arquitectura de la solución

Se plantea una solución desplegada en la nube por parte del backend y desplegada en la play store por parte del móvil, asegurando la disponibilidad y rendimiento de la aplicación. Se cuenta con un API REST al cual la aplicación móvil se conecta con los servicios del backend y procesa la información por medio de objetos JSON.

### Tecnologías utilizadas

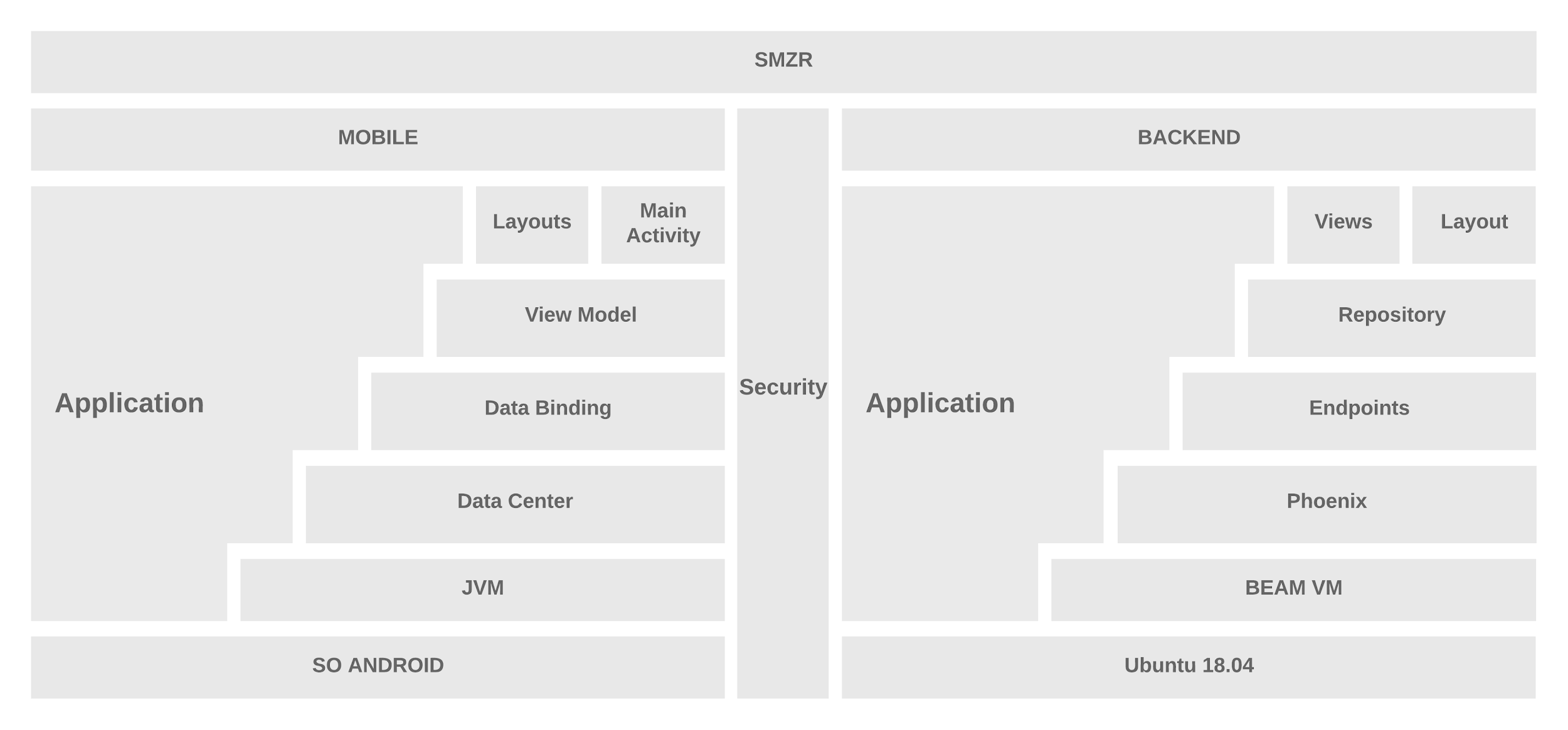
* Android Studio
* Kotlin
* Retrofit
* SQLite
* PostgreSQL
* Elixir
* JWT
* Amazon RDS
* Amazon EC2
* Gradle
* Firebase
* Google maps Service



## **Vistas de la solución**

### Vista modular

En esta vista se muestra cómo el sistema va a estar estructurado, este consta de dos partes, una en mobile y la otra en backend, en la estructura modular del móvil tenemos la capa principal de vista las cuales interactúan sólo con el view model para evitar problemas de rendimiento, el view model se encargará de procesar la data mediante el data binding, esta data se comunicará con los distintos data center que tenemos (local sqlite y externa con postgresql), será compilado por la JVM en un sistema operativo android. En la estructura modular del backend tenemos los views y layout los cuales procesan los archivos JSON, luego tenemos el repository que es donde se interactúa con la base de datos postgres, todos estos resultados son devueltos por medio de endpoints hacia la aplicación android las cuales consume.



### **Vista de despliegue**

En esta vista las cajas tridimensionales, conocidas como nodos, representan los elementos básicos de software o hardware, o nodos, en el sistema. Las líneas de nodo a nodo indican relaciones, y las formas más pequeñas contenidas dentro de las cajas representan los artefactos de software que se implementan.

Los nodos pueden ser de dos tipos: Nodos del dispositivo (como PC, computadoras portátiles o teléfonos móviles) y nodos del entorno de ejecución (como un sistema operativo, una JVM u otro contenedor de servlets). Para el módulo de Call Center hay nodos del dispositivo como PC del operador y PC del supervisor de Call Center. Además hay nodos del entorno de ejecución como el sistema operativo Ubuntu Server 18.04 LTS. El artefacto es dbSchema.