**Universidad Técnica Particular de Loja**

**Sistemas Informáticos y Computación**

**Integrantes:**

* Kevin José Quito Medina
* Freddy Stalin Villavicencio Espinoza

**Asignatura:** Sistemas Basados en Conocimiento

**Docente:** Ing. Janneth Alexandra Chicaiza Espinosa

**INTRODUCCIÓN**

RDF es un método para expresar el conocimiento en un mundo descentralizado y es el fundamento de la Web Semántica, en el que las aplicaciones informáticas utilizan información estructurada distribuida por toda la Web. Proporciona interoperabilidad entre aplicaciones que intercambian información legible por máquina en la Web. RDF destaca por la facilidad para habilitar el procesamiento automatizado de los recursos Web.

RDF puede utilizarse en distintas áreas de aplicación: como en el presente trabajo, en donde se realizo un proceso de obtención y limpieza de datos acerca de los casos de COVID-19 en el continente Africano para la posterior generación y almacenamiento de datos RDF, culminando con el desarrollo de una aplicación web en donde se evidencia los resultados obtenidos.

El siguiente informe está estructurado de acuerdo con las etapas del proyecto de la siguiente manera:

* Obtención de datos: En esta etapa se describen las fuentes de donde se tomaron los datos para la posterior generación de datos RDF.
* Transformación de datos RDF: Aquí se describen las tareas de limpieza, selección de patrones URIs, la lógica de transformación de datos con JENA y los resultados de la transformación. Además de indicar el repositorio elegido para el almacenamiento de datos RDF.
* Aplicación: En este apartado se describen los trabajos relacionados, además de la arquitectura y tecnologías empleadas para el desarrollo de la aplicación.
* Conclusiones: en esta etapa se indican las conclusiones del proyecto.

**DESARROLLO**

1. **OBTENCIÓN DE DATOS**

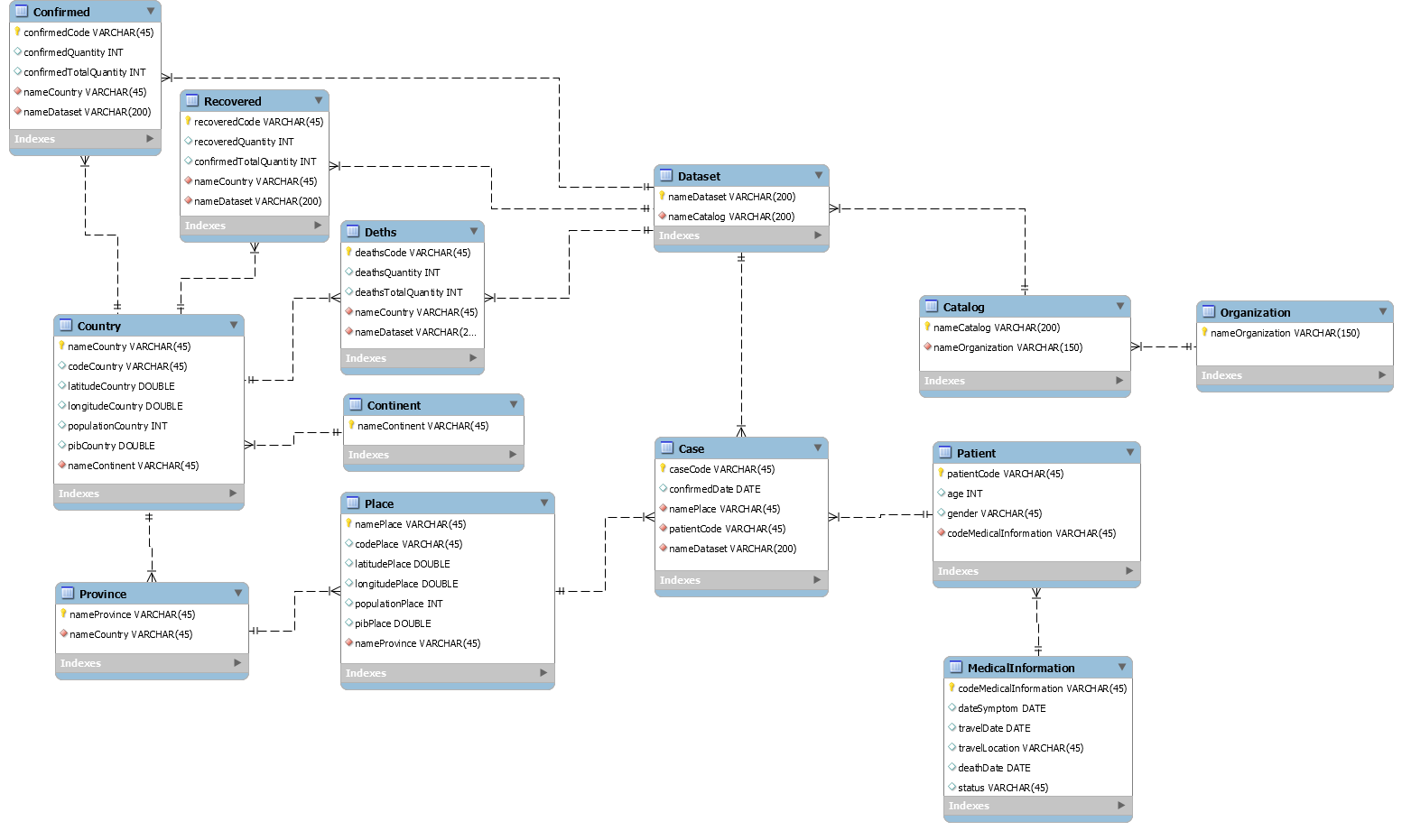
En el presente proyecto se trabajaron con dos fuentes de datos, que se describen a continuación:

|  |  |  |  |
| --- | --- | --- | --- |
| **Fuente** | **URL** | **Formato de datos** | **Actualización** |
| COVID-19 África | http://covid-19-africa.sen.ovh/index.php?isnc=2 | CSV | Actualización diaria |
| Coronavirus COVID-19 (2019-nCoV) Data Repository for Africa | https://github.com/dsfsi/covid19africa | CSV | Actualización hace un mes |

La fuente **COVID-19 África** nos brinda información acerca de estadísticas de casos confirmados, recuperados y fallecidos de cada uno de los países de África. La fuente **Coronavirus COVID-19 (2019-nCov) Data Repository for África** nos ofrece información sobre los casos confirmados (Género, Fecha de confirmación, Edad, etc). Realizamos la descarga de la data en formato CSV desde las fuentes mencionadas. La data se encuentra actualizada hasta el día 16/06/2020.

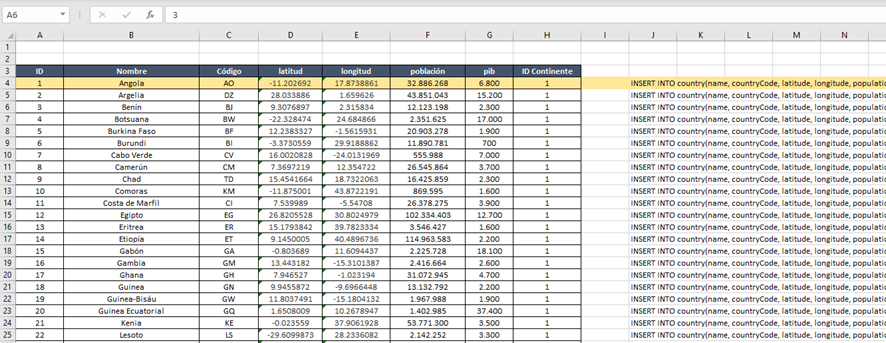
1. **OBTENCIÓN DE DATOS RDF**
   1. **Tareas de limpieza**

En base al modelo ontológico definido, se estableció un modelo de base de datos relacional que se adapta a las fuentes de datos recolectadas mencionadas anteriormente (Figura. 1).



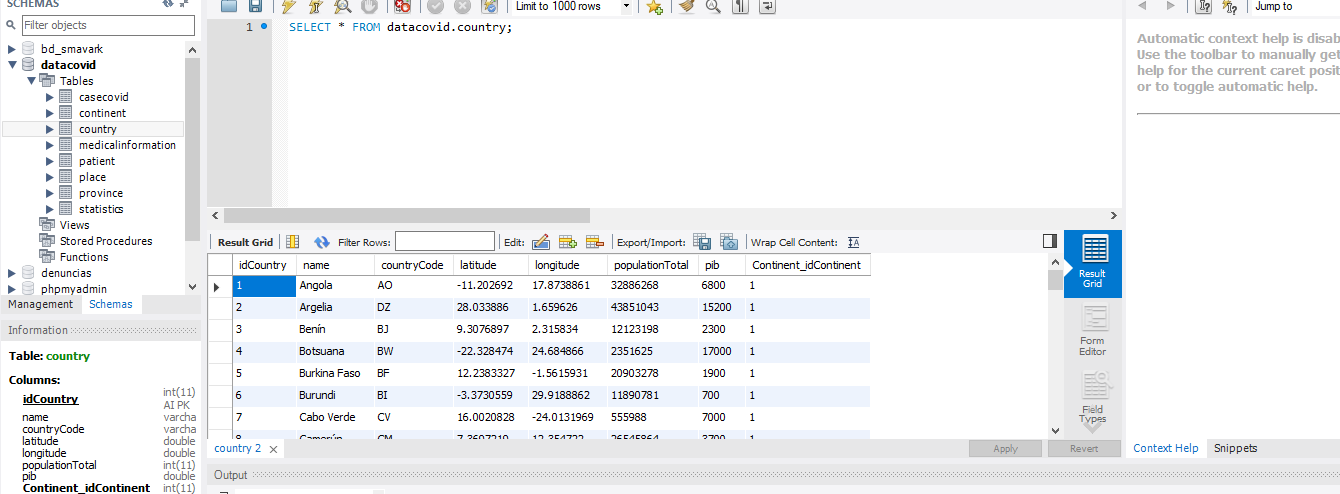
*Figura. 1. Modelo relacional de base de datos.*

A continuación, se realizó la limpieza de los datos, separándolo por entidades con sus respectivos atributos y relaciones. Además, se realizó el cambio en el formato de las fechas. A sí mismo, se generaron los INSERT SQL de forma automática. Estas tareas se las realizó en Excel (Figura. 2).



*Figura. 2. Preparación de datos*

Finalmente se realizó la inserción de los datos en una base de datos MySQL. (Figura. 3).



*Figura. 3. Datos almacenados en la base de datos MySQL*

* 1. **Selección de patrones URI y definición de licencia**

Para la definición de las diferentes URIs optamos por utilizar URIs significativos y barras URI (303-ASCII) como se menciona en la guía de referencia para realizar la especificación de URIs. Obteniendo así los siguientes elementos URI:

* Estructura base de URI: Para la iniciativa referente a la data de COVID-19 de Linked Data hemos definido el siguiente dominio.

http://utpl.edu.ec/lod/dataCOVID/

* URI de TBox: Anexamos el nombre del concepto o de la propiedad a la estructura base del URI para incluir conceptos y propiedades disponibles en nuestra ontología.

http://utpl.edu.ec/lod/dataCOVID/ontology/{concepto o propiedad}

* URI de ABox: Anexamos el nombre del recurso a la estructura base del URI para obtener la información de una instancia de un recurso.

http://utpl.edu.ec/lod/dataCOVID/resource/{recurso}

Para poder identificar instancias de cada tipo de recurso, se utilizó las siguientes normas:

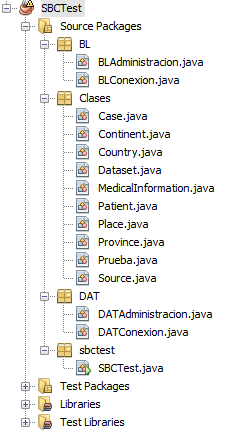
|  |  |  |
| --- | --- | --- |
| **Clase** | **Identificador** | **Ejemplo** |
| Continent | <Nombre del continente> | /África |
| Country | <Nombre del país> | /Angola |
| Province | <Nombre de la provincia> | /Luanda |
| Place | <Nombre del lugar> | /Adrar |
| Organization | <Nombre de la organización> | /MSPE |
| Catalog | <Nombre del catálogo> | /MSPE-covid19 |
| Dataset | <Nombre del dataset> | /Covid19-Africa |
| MedicalInformation | <”Information-” + número de 6 dígitos> | /Information-000001 |
| Patient | <”Patient-” + número de 6 dígitos> | /Patient-000001 |
| CaseCovid | <”Case-” + número de 6 dígitos> | /Case-000001 |
| Confirmed\_cases | <”Confirmed-” + número de 6 dígitos > | /Confirmed-000001 |
| Deaths\_cases | <”Deaths-” + número de 6 dígitos > | /Deaths-000001 |
| Recovered\_cases | <”Recovered-” + número de 6 dígitos > | /Recovered-000001 |

El proyecto se desarrolló bajo licencia ***CREATIVE COMMONS*** puesto que nos permite la distribución de obras con derechos de autor. Cabe mencionar que las fuentes de datos que definimos para el proyecto cuentan con la misma licencia lo que permite reutilizar los recursos especificados.

Entre otras cosas, esta licencia permite que otros distribuyan, mezclen, adapten y desarrollen su trabajo, incluso comercialmente, siempre y cuando proporcionen el crédito por la creación original. Esta es la más complaciente de las licencias ofrecidas y es recomendado para la máxima difusión y uso de materiales con licencia.

* 1. **Lógica de Transformación de datos con JENA**

Para la transformación de los datos se aplicó una lógica y arquitectura en 3 capas, en la cual se encuentra inicialmente la capa de datos que se encarga de gestionar todo lo relativo a la base de datos y a la creación, edición y borrado de datos de ésta.

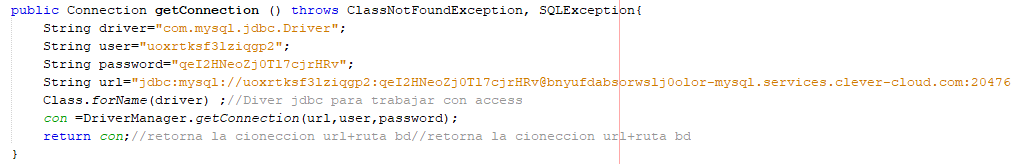
Luego se realiza la conexión con la siguiente capa que es la capa de negocio en donde se gestiona la lógica de la aplicación. Es donde se dice que se hace con los datos y esta capa estará conectada con la capa de persistencia para poder realizar sus funciones.

Y finalmente en la capa de presentación contamos con una clase en donde realizamos en base a las consultas de base de datos la transformación de estos a formato RDF con el framework JENA.

En la imagen presentada anteriormente en el paquete Clases, se encuentran varias clases .java creadas en base a nuestra ontología sobre datos COVID-19.

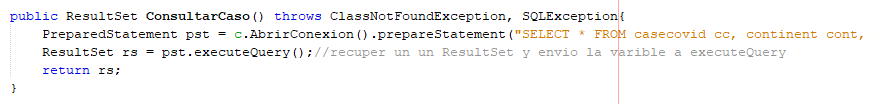
A continuación, se realizará la explicación breve del proceso que realizamos para llegar a la transformación de datos:

Primeramente, realizamos la conexión a la base de datos llamada ***datacovid*** como se muestra en la figura 4.



*Figura 4. Conexión a la base de datos de MySQL*

Luego realizamos una consulta multi tabla para obtener todos los datos relacionados a un caso de COVID-19 este proceso se muestra en la figura 5.



*Figura 5. Consulta a la base de datos*

Finalmente, como se muestra en la figura 6, procesamos el resultado de la consulta anterior instanciando los objetos de las clases .java para poder crear un objeto de un caso de covid y almacenarlo en un arraylist del mismo tipo para posteriormente poder manipularlo y crear tripletas RDF en base al mismo.



*Figura 6. Procesamiento de consulta de base de datos*

En el arraylist almacenamos temporalmente el resultado de la consulta de la base de datos anteriormente mencionada.

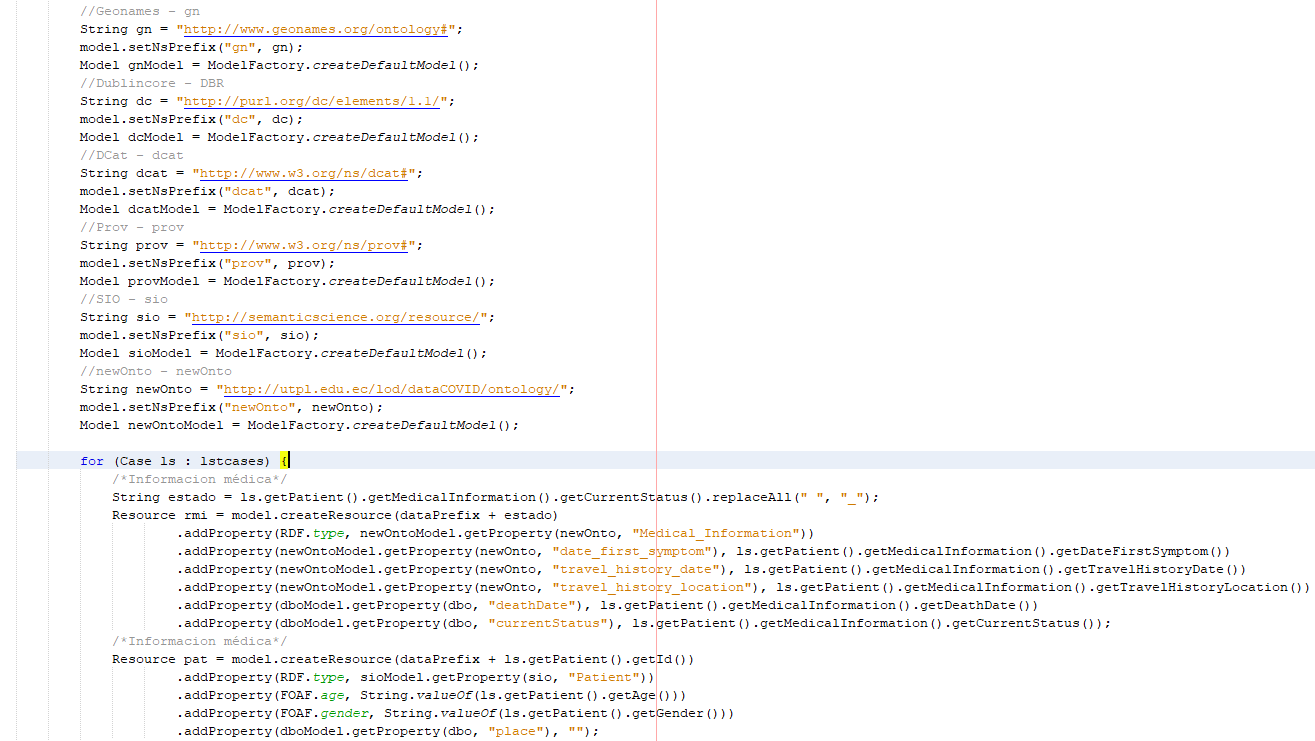


Para creación de las diferentes propiedades que corresponden al modelo ontológico definimos los prefijos necesarios como se muestra en la figura 7.



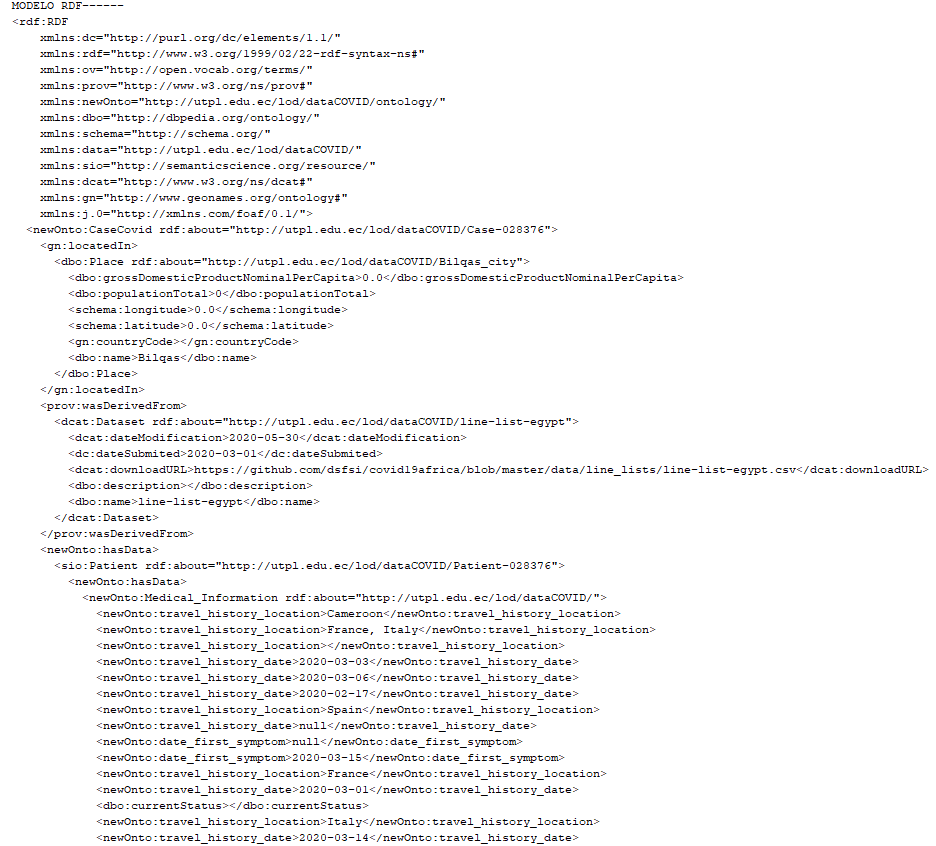
*Figura 7. Definición de prefijos*

A continuación, se presenta la reutilización los métodos proporcionados por el docente para la creación de datos RDF con los datos del arraylist como se muestra a continuación en la figura 8.



*Figura 8. Arraylist para creación de datos RDF*

Al ejecutar el código anterior se generan las tripletas RDF de cada caso de COVID-19 con sus respectivos datos y relaciones como se muestra a continuación en la figura 9.



*Figura 9. Resultado de la ejecución del código para la generación de datos RDF*

* 1. **Resultados de la transformación**

A continuación, se presenta el número de instancias que se generaron a partir de los datos recolectados para cada clase del modelo ontológico creado.

|  |  |
| --- | --- |
| **Clase** | **Número de instancias** |
| Continent | 1 |
| Country | 54 |
| Province | 123 |
| Place | 150 |
| Organization | 1 |
| Catalog | 2 |
| Dataset | 47 |
| MedicalInformation | 69023 |
| Patient | 69023 |
| CaseCovid | 69023 |
| Confirmed\_cases | 3388 |
| Deaths\_cases | 1276 |
| Recovered\_cases | 2263 |

* 1. **Elección de repositorio de almacenamiento**



Elegimos **VIRTUOSO** porque tiene una capacidad de virtualización de datos que permite la construcción e implementación de grafos de conocimiento sobre datos existentes expuestos por APIS como HTTP, ODBC, JDBC, entre otros.

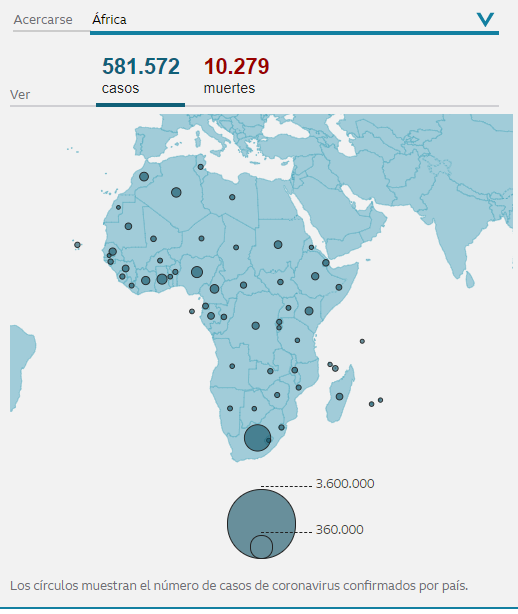
También cuenta con una base de datos MySQL que permite almacenar información para la creación de diferentes grafos. Además, tiene la función de importar y almacenar datos RDF directamente para la posterior utilización en forma de grafo.

Virtuoso ofrece el servicio de ***Virtuoso Conductor*** que facilita la administración de la base de datos, de los grafos, de las vistas para grafos, tablas, procedimientos almacenados, copias de seguridad, consultas Sparql, administración de dominios locales y directorios.

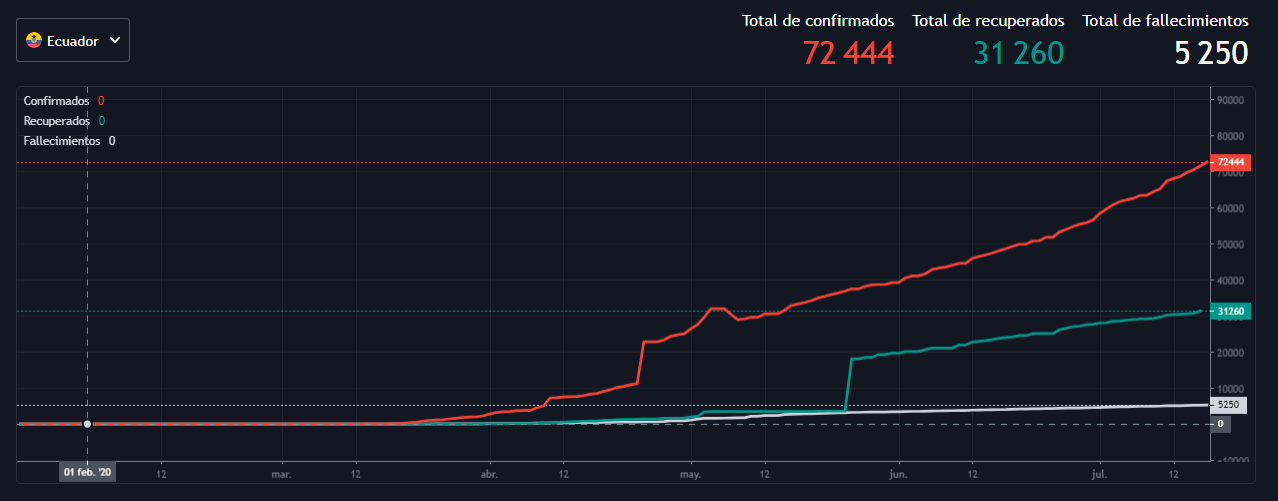
1. **Aplicación**
   1. **Trabajos relacionados**

**CORONAVIRUS MAPEADO:** Los casos confirmados por país son representados en un mapa mediante círculos de tamaño proporcional al número de casos confirmados.

Este trabajo fue desarrollado por la Universidad Johns Hopkins (Baltimore, EE.UU.), y autoridades locales. La última actualización de estas cifras se realizó el 3 de julio de 2020.



**GRÁFICOS Y ESTADÍSTICAS DEL CORONAVIRUS (COVID - 19):** Representa mediante una línea de tiempo la variación de las cifras respecto a los casos confirmados, recuperados y fallecidos a nivel mundial y por país.



* 1. **Herramientas utilizadas**

**Virtuoso:** Virtuoso Universal Server es un híbrido de middleware y motor de base de datos que combina la funcionalidad de un sistema tradicional de gestión de bases de datos relacionales (RDBMS), base de datos relacional de objetos (ORDBMS), base de datos virtual, RDF , XML , texto libre , servidor de aplicaciones web y servidor de archivos funcionalidad en un solo sistema.

**Spring Boot:** Spring es un framework alternativo al stack de tecnologías estándar en aplicaciones JavaEE.

**Dependencia Jena de Spring Boot:** Apache Maven es una herramienta para ayudar a los proyectos Java a administrar sus dependencias en el código de la biblioteca, como Jena. Al declarar una dependencia en el núcleo de Jena en el pom.xml archivo de su proyecto, obtendrá también el conjunto consistente de archivos de biblioteca de los que Jena depende.

**Angular:** Angular es un framework opensource desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application).

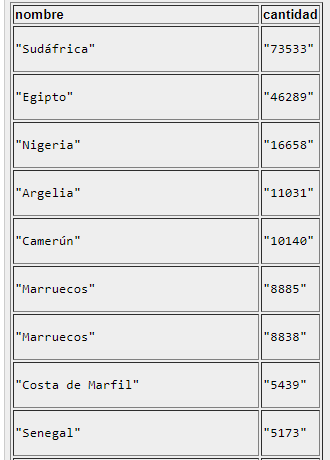
* 1. **Consultas Sparql**

Se realizaron las siguientes consultas con el fin de dar una respuesta a las preguntas establecidas en la definición del proyecto.

**Consulta 1:** ¿Qué países cuentan con el mayor número de casos confirmados con covid-19 hasta la última fecha de actualización de datos?

|  |
| --- |
| PREFIX newOnto:<http://utpl.edu.ec/lod/dataCOVID/ontology/>  PREFIX gn:<http://www.geonames.org/ontology#>  PREFIX schema:<http://schema.org/>  SELECT ?nombre ?cantidad WHERE {  ?var rdf:type newOnto:Confirmed\_Cases ;  newOnto:totalQuantity ?cantidad ;  gn:locatedIn ?pais;  schema:observationDate ?fecha.  ?pais dbo:name ?nombre.  FILTER (?fecha = "2020-06-15")  }ORDER BY DESC (xsd:integer(?cantidad)) |

**Resultado de la consulta 1:**



**Consulta 2:** ¿Cuál es el género con mayor número de casos de covid-19?

|  |
| --- |
| PREFIX sio:<http://semanticscience.org/resource/>  SELECT (sum(if($gender="male",1,0)) as ?hombres)  (sum(if($gender="female",1,0)) as ?mujeres)  {  {SELECT DISTINCT ?person $gender WHERE {  ?person rdf:type sio:Patient;  foaf:gender $gender  } GROUP BY ?person}  } |

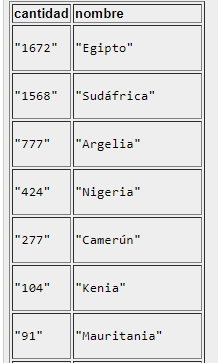
**Resultado de la consulta 2:**

****

**Consulta 3:** ¿Qué países presentan el mayor índice de casos fallecidos por covid-19?

|  |
| --- |
| PREFIX newOnto:<http://utpl.edu.ec/lod/dataCOVID/ontology/>  PREFIX gn:<http://www.geonames.org/ontology#>  PREFIX schema:<http://schema.org/>  SELECT ?cantidad ?nombre WHERE {  ?var rdf:type newOnto:Deaths\_Cases ;  newOnto:totalQuantity ?cantidad ;  gn:locatedIn ?pais;  schema:observationDate ?fecha.  ?pais dbo:name ?nombre.  FILTER (?fecha = "2020-06-15")  }ORDER BY DESC (xsd:integer(?cantidad)) |

**Resultado de la consulta 3:**

****

**Consulta 4:** ¿Qué países visitaron las personas contagiadas con COVID-19 antes de ingresar al continente africano?

|  |
| --- |
| PREFIX newOnto:<http://utpl.edu.ec/lod/dataCOVID/ontology/>  PREFIX gn:<http://www.geonames.org/ontology#>  PREFIX schema:<http://schema.org/>  SELECT ?case ?location WHERE {  ?case rdf:type newOnto:CaseCovid;  newOnto:hasData ?paciente.  ?paciente newOnto:hasData ?informacion.  ?informacion newOnto:travel\_history\_location ?location.  } |

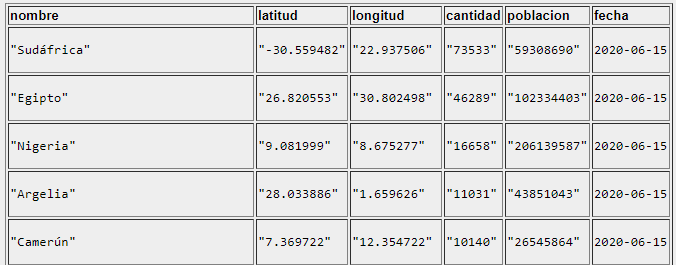
**Resultado de la consulta 4:**



**Consulta 5:** ¿Cuál el nombre de los países, su longitud, su latitud, su población y el número de casos de covid-19?

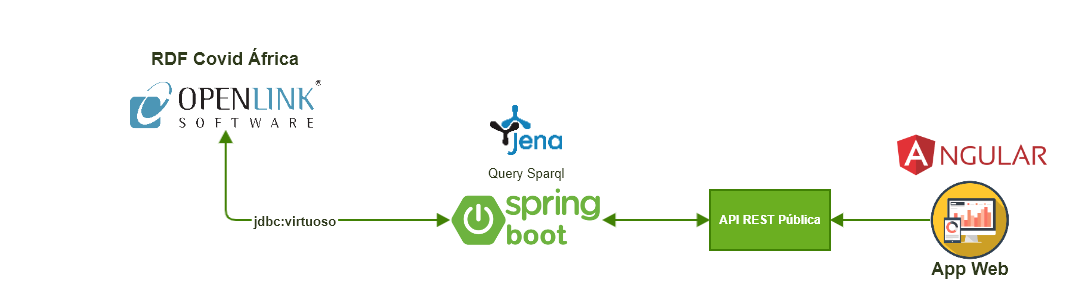
|  |
| --- |
| PREFIX newOnto:<http://utpl.edu.ec/lod/dataCOVID/ontology/>  PREFIX gn:<http://www.geonames.org/ontology#>  PREFIX schema:<http://schema.org/>  PREFIX dbo: <http://dbpedia.org/ontology/>  SELECT ?nombre ?latitud ?longitud ?cantidad ?poblacion ?fecha WHERE {  ?var rdf:type newOnto:Confirmed\_Cases ;  newOnto:totalQuantity ?cantidad ;  gn:locatedIn ?pais;  schema:observationDate ?fecha.  ?pais dbo:name ?nombre;  schema:latitude ?latitud;  schema:longitude ?longitud;  dbo:populationTotal ?poblacion.  FILTER (?fecha = "2020-06-15")  }ORDER BY DESC (xsd:integer(?cantidad)) |

**Resultado de la consulta 5:**



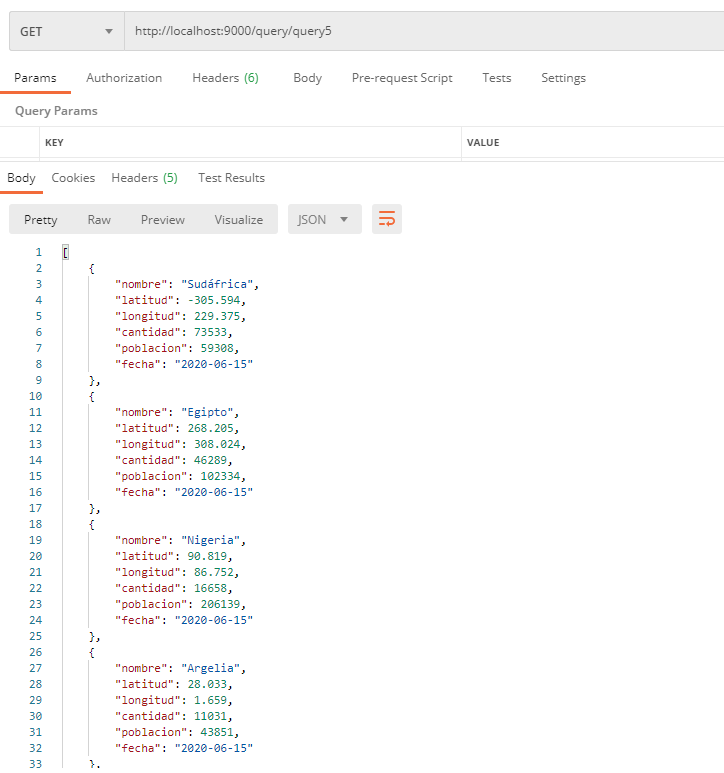
* 1. **Lógica de la app**

A continuación, se describe la arquitectura de la aplicación:



En un principio se cuenta con el rdf (Covid-19 África) generado en Jena en la fase anterior del proyecto. Este rdf se encuentra almacenado en Virtuoso OpenLink Software y se lo consume desde Spring Boot (Framework JAVA) mediante conexión con el jdbc:virtuoso para realizar consultas Sparql para la posterior exposición de los resultados de esta mediante una API REST Pública para finalmente ser consumida desde cualquier aplicación.

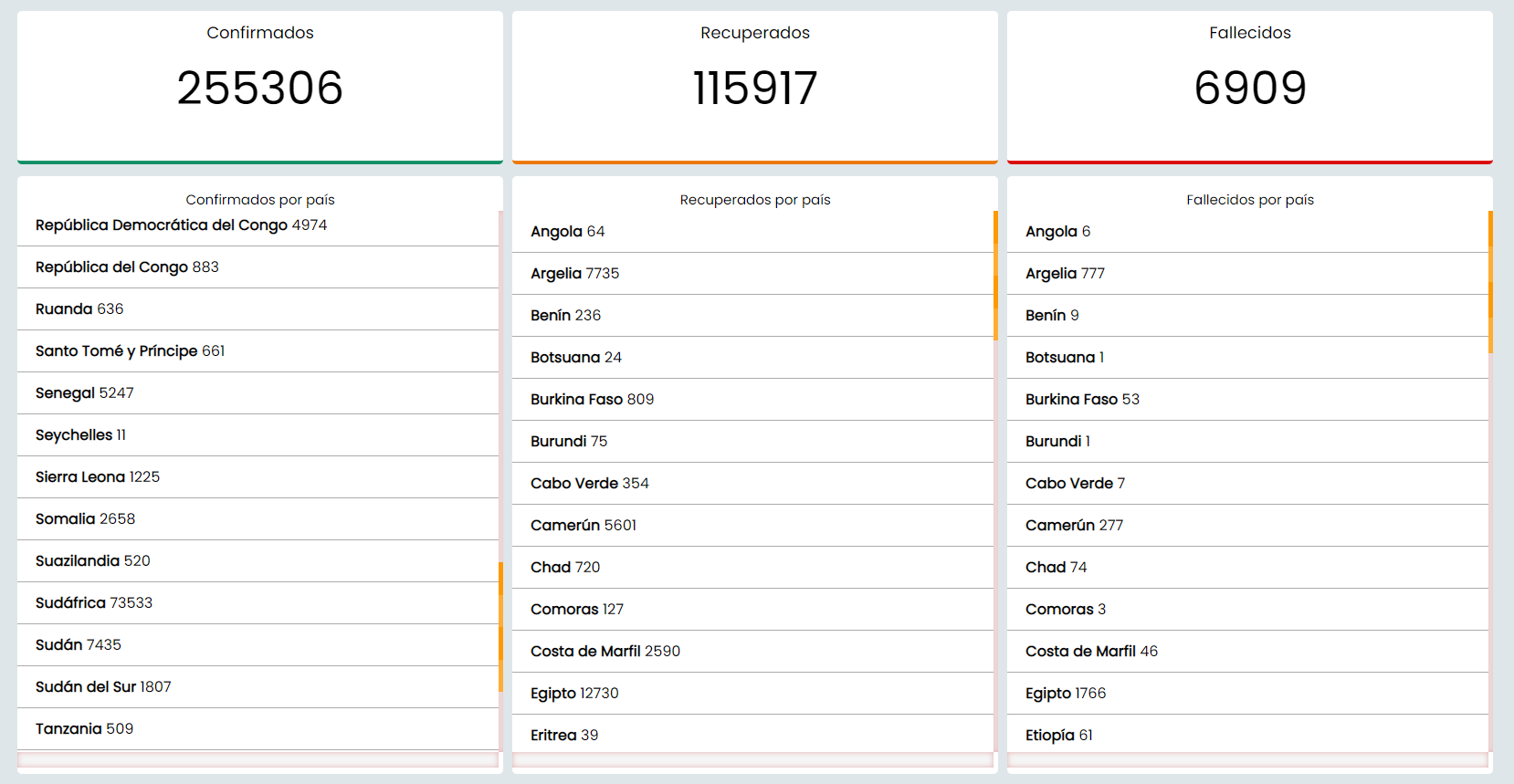
La API generada con los resultados de una consulta Sparql tiene la siguiente estructura:



Las APIs generadas se las consumen desde el Front-end de la aplicación:

En la página inicial de la aplicación se puede visualizar una breve descripción de la situación de COVID-19 en África, seguido del número de casos confirmados, recuperados y fallecidos y tablas en donde se muestra estos datos por país.

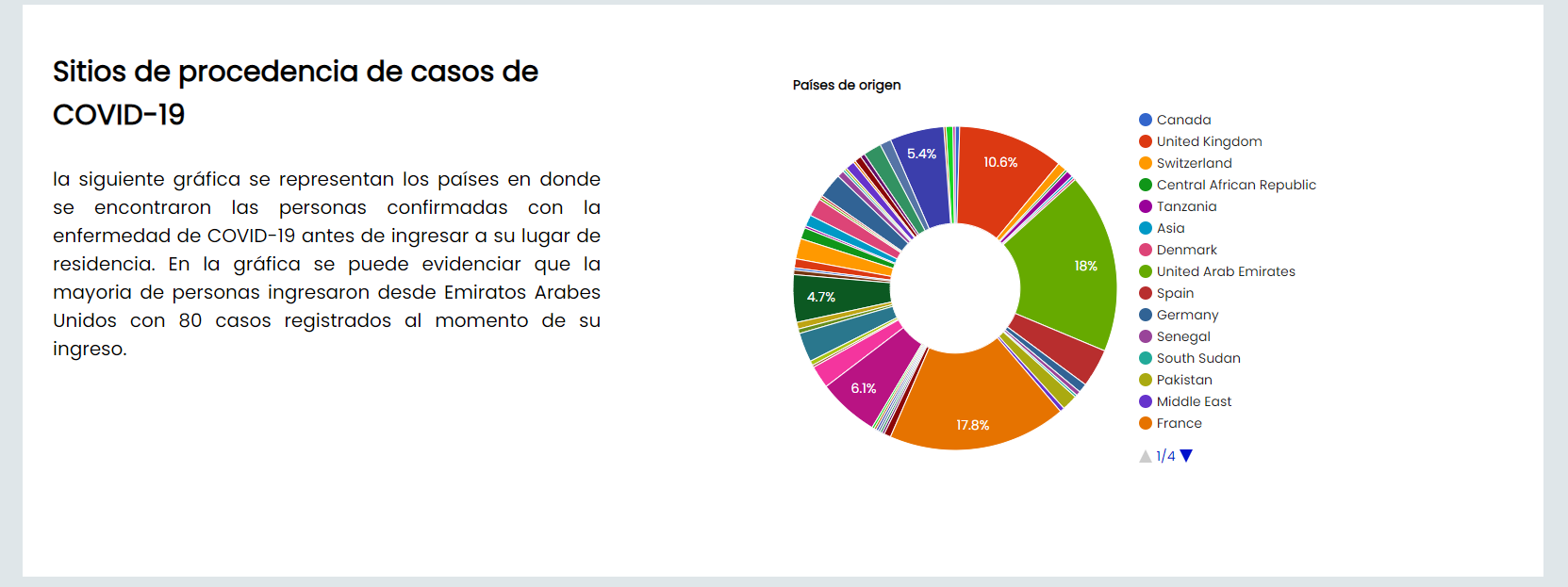




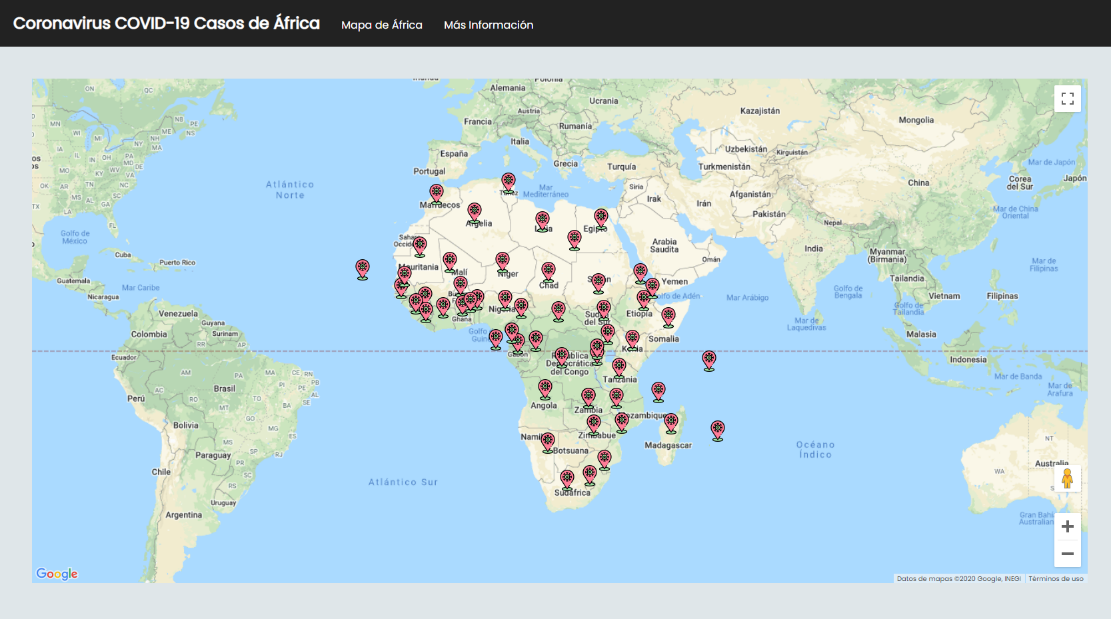
En la siguiente gráfica de barras se puede visualizar el número de casos por paí.



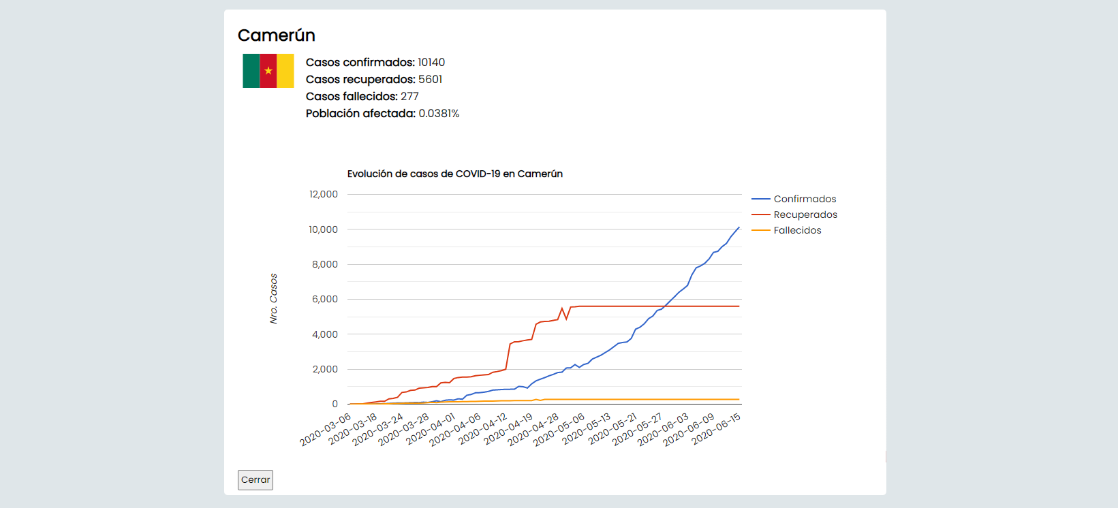
En la siguiente gráfica se muestra los sitios de donde ingresaron las personas con COVID-19.



En el mapa se graficaron indicadores en cada uno de los países afectados por COVID-19 en el continente africano.



Al presionar un marcador del mapa se no abre una ventana en donde podemos visualizar información sobre dicho país: casos confirmados, recuperados, fallecidos, porcentaje de la población afectada y una línea de tiempo en donde se puede visualizar la evolución diaria de la enfermedad en ese país.



1. **Conclusiones**

El proyecto de generación de datos RDF nos permite evidenciar todo el proceso realizado para llegar a la creación de la aplicación web que contiene todos los datos recopilados y almacenados en el repositorio de datos especificado con anterioridad.

Después de la realización del actual proyecto llegamos a la conclusión de que el proceso realizado con todas la etapas y características aplicadas a cada una de ellas nos permite en cierta parte alcanzar el objetivo de la web semántica de la publicación de datos legibles para aplicaciones informáticas como en nuestro caso la publicación y realización de la aplicación de visualización de datos estadísticos sobre los casos de COVID-19 en el continente africano.