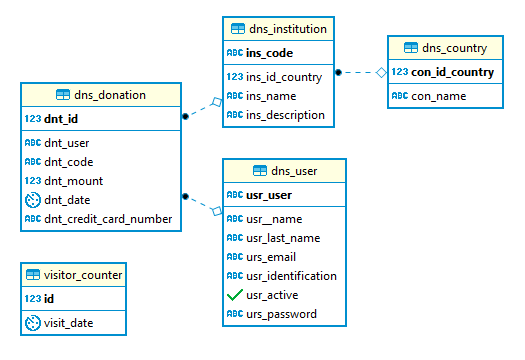
# **Donation Site [Backend]**

**Nombre: Ricardo Salvador Rivas Franco.**

1. **Modelo entidad relación:**

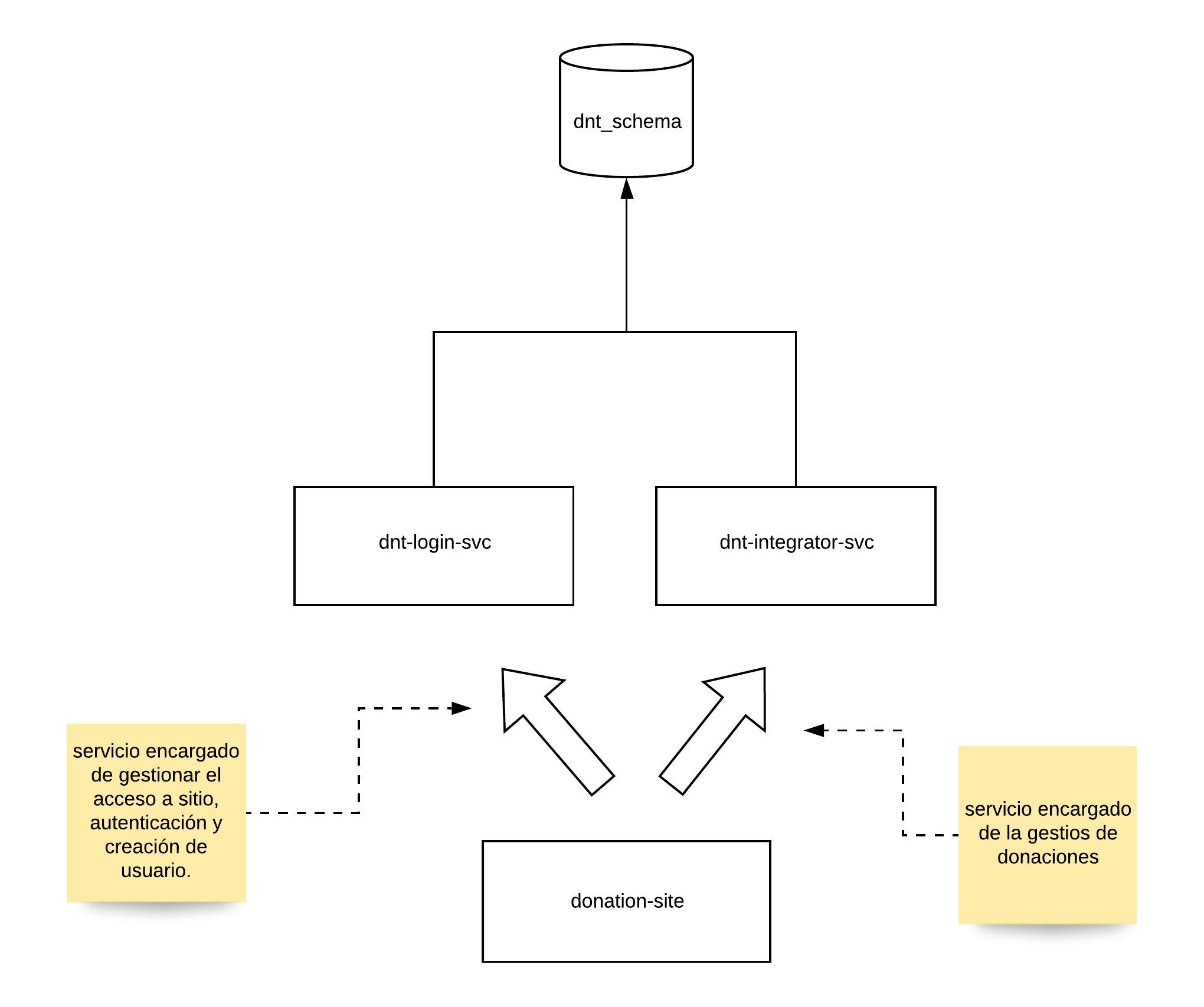


El diseño de modelo entidad relación se conforma de la siguiente manera:

* **DNS\_DONATION:** Tabla que contiene la información de las donaciones, esta involucra al usuario, la institución beneficiadas, el monto y la tarjeta utilizada para la transacción y la fecha realizadas.
* **DNS\_USER:** Tabla que almacena información relacionada al usuario: correo electrónico, nombre, apellidos, identificación y contraseña. Además, un campo usr\_active de tipo Boolean para definir si la cuenta esta activa o bloqueada.
* **DNS\_INSTITUTION:** En esta tabla se guarda la información de las instituciones beneficiadas con las donaciones de los usuarios, se almacena el nombre, código único de identificación y país al que pertenece.
* **DNS\_COUNTRY:** contiene el catálogo de países.
* **VISITOR\_COUNTER:** Tabla que almacena la fecha de cada visita.

1. **Solución arquitectónica:** El aplicativo está conformado de la siguiente manera

* **dnt-login-svc**: Servicio encargado de gestionar el acceso a la aplicación, además de agregar nuevos usuarios. Utiliza Spring Security para un mejor control de acceso al servicio.
* **dnt-integrator-svc:** Servicio encargado de gestionar las donaciones y de consultar las visitas realizadas.

****

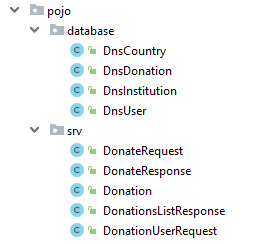
1. **Descripción del código fuente:**

Los servicios se construyeron utilizando Java 8, Framework Spring Boot 2, JPA para la conexión de base de datos, Base de datos PostgreSQL 9.X, Apache Tomcat, Spring Security (dnt-login-svc) para el control de acceso al servicio. El código fuente se conforma de los siguientes elementos:

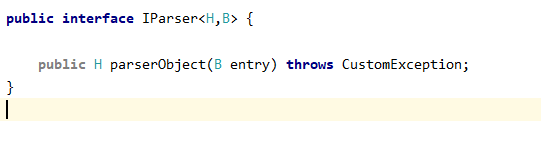
* **Controller**: En estas clases se definen los puntos de accesos, parámetros de entrada y el método de llegar a ellos (POST, GET, PUT, etc.) 
* **Process**: Clase encargada de llevar el flujo principal del negocio, es decir, define el paso a paso el procesamiento de la información ingresada sin conocer cómo se lo hacen cumpliendo así el principio de Inversión de dependencias.



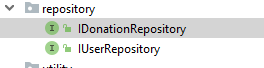
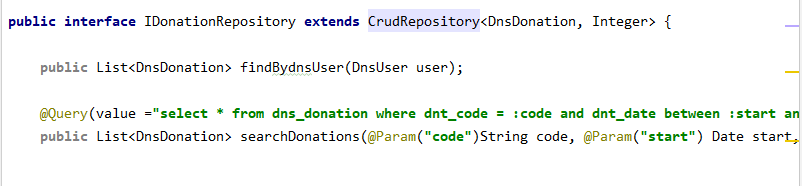
* **Pojos**: Paquete que contiene todas las clases del negocio, desde objetos de base de datos hasta contratos de entradas y salidas de los servicios. Estas clases no poseen lógica ni implementan nada en especial.



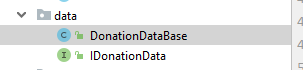
**Parser:** Estas clases se utilizan para obtener un objeto de una clase a partir de otro objeto u objetos, Para ello se hace siempre una implementación de una interfaz para acomodarla a la situación sin cambiar su idea principal, cumpliendo el principio de Abierto/Cerrado.



**Repository:** En este paquete se almacenan las interfaces encargadas de interactuar con la base de datos, extienden de la librería CrudRepository. Permite el uso de funciones preestablecidas (Save, Delete, Update, etc.) o bien permite hacer uso de queries nativas.

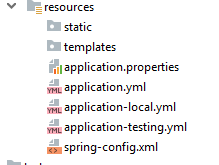
**Data**: En esta paquetería se crean todas aquellas clases que acceden una fuente de datos externas, como una base de datos, API, servicios SOAP, etc.

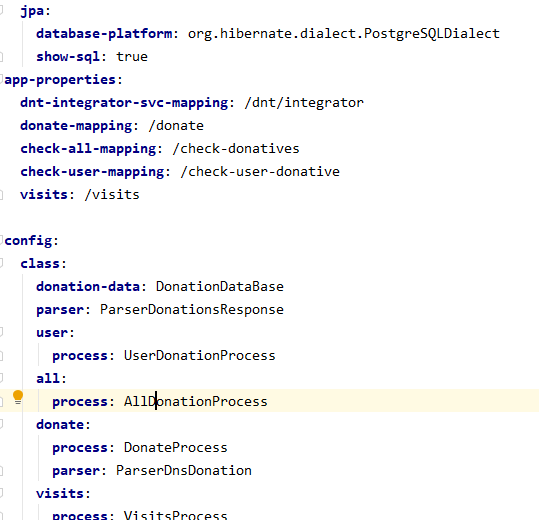


**Configurations:** Guarda todas aquellas clases de configuración que el servicio necesita para un despliegue correcto, en ella se colocan beans, instanciación de objetos, configuración de seguridad para el caso en que se utiliza Spring Security.



**Resources**: En esta carpeta almacena todos los archivos de configuración, la cual contiene variables globales que serán utilizadas por diferentes clases, además, posee los datos necesarios para la conexión a la base de datos y la definición de URLs de los puntos de acceso.





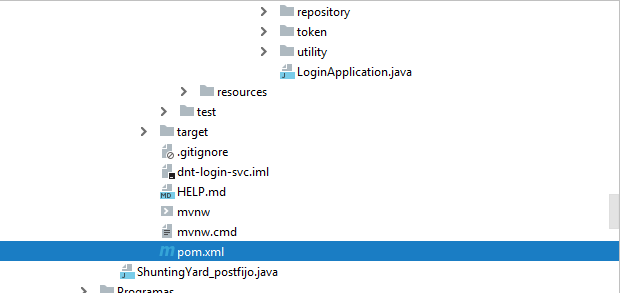
1. **Descripción de endpoint:**

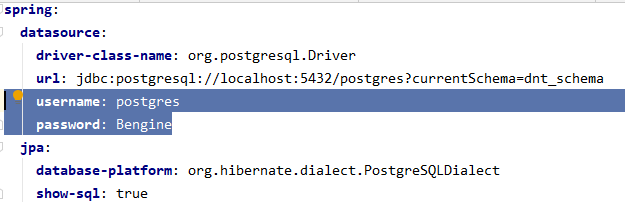
|  |
| --- |
| **dnt-integrator-svc** |
| **URL:** localhost:90/dnt/integrator/check-user-donative [POST] |
| **Descripción:** Obtiene todos los donativos realizados por un usuario en específico. |
| **REQUEST:** |
| **{**  **"user": "rrivas"**  **}** |
| **RESPONSE:** |
| {      "donations": [          {              "Surnames": "Rivas",              "Name": "Ricardo",              "id\_document": "019223232",              "country": "Colombia",              "institution": "Asociación de Academias de la Lengua Española",              "donation\_amount": "12.00",              "date": "2019-10-12"          },          {              "Surnames": "Rivas",              "Name": "Ricardo",              "id\_document": "019223232",              "country": "El Salvador",              "institution": "Plan International",              "donation\_amount": "130.00",              "date": "2019-12-12"          }      ]  } |
| **URL:** localhost:90/dnt/integrator/check-donatives [GET] |
| **Descripción:** Obtiene todos los donativos realizados. |
| **RESPONSE:** |
| {      "donations": [          {              "Surnames": "Rivas",              "Name": "Ricardo",              "id\_document": "019223232",              "country": "Colombia",              "institution": "Asociación de Academias de la Lengua Española",              "donation\_amount": "12.00",              "date": "2019-10-12"          },          {              "Surnames": "Rodriguez",              "Name": "Albeto",              "id\_document": "1123434343",              "country": "Colombia",              "institution": "Asociación de Academias de la Lengua Española",              "donation\_amount": "5000.00",              "date": "2019-03-02"        ]  } |
| **URL:** localhost:90/dnt/integrator/donate [POST] |
| **Descripción:** Endpoint para realizar una donación. |
| **REQUEST:** |
| **{**  **"user": "rrivas",**  **"institution\_cod": "ASALE",**  **"mount": 1000,**  **"credit\_card": "4508200012111121"**  **}** |
| **RESPONSE:** |
| {      "status": true,      "messages": ""  } |
| **URL:** localhost:90/dnt/integrator/visits [GET] |
| **Descripción:** Retorna la cantidad de veces que ha sido consultado los endpoint /check-donatives y /check-user-donative |
| **RESPONSE:** |
| {      "overall\_visits": 12  } |
| **dnt-login-svc** |
| **URL:** http://localhost:90/authentication/adduser [POST] |
| **Descripción:** Agrega un nuevo usuario a la base de datos. |
| **REQUEST:** **{**  **"user": "cavila8",**  **"name": "Carlos",**  **"surnames": "Aguilar",**  **"email": "caguilar@correo.com",**  **"identification": "1233",**  **"pass": "n1SLFI36W444OP7G+AruwA=="**  **}** |
| **RESPONSE:** |
| {      "headers": {},      "body": {          "status": true,          "message": ""      },      "statusCodeValue": 200,      "statusCode": "OK"  } |
| **URL:** http://localhost:90/authentication/login [POST] |
| **Descripción:** Endpoint de autenticación a un usuario. |
| **REQUEST:** |
| {  "user": "rrivas",  "password": "n1SLFI36W444OP7G+AruwA=="  } |
| **RESPONSE**: |
| {      "headers": {},      "body": {          "status": true,          "message": ""      },      "statusCode": "OK",      "statusCodeValue": 200  } |

1. **Instalación**

A continuación, se detalla los pasos para ejecutar el proyecto, estos se aplican para ambos servicios dnt-login-svc y dnt-integrator-svc:

1. Restaurar el archivo **dump3-postgres-201912230535.backup** en base de datos PostgreSQL 9.X.
2. Descarga y descomprimir proyectos
3. Abrir IDE, localizar proyecto y abrir desde el archivo ../login/POM.xml.



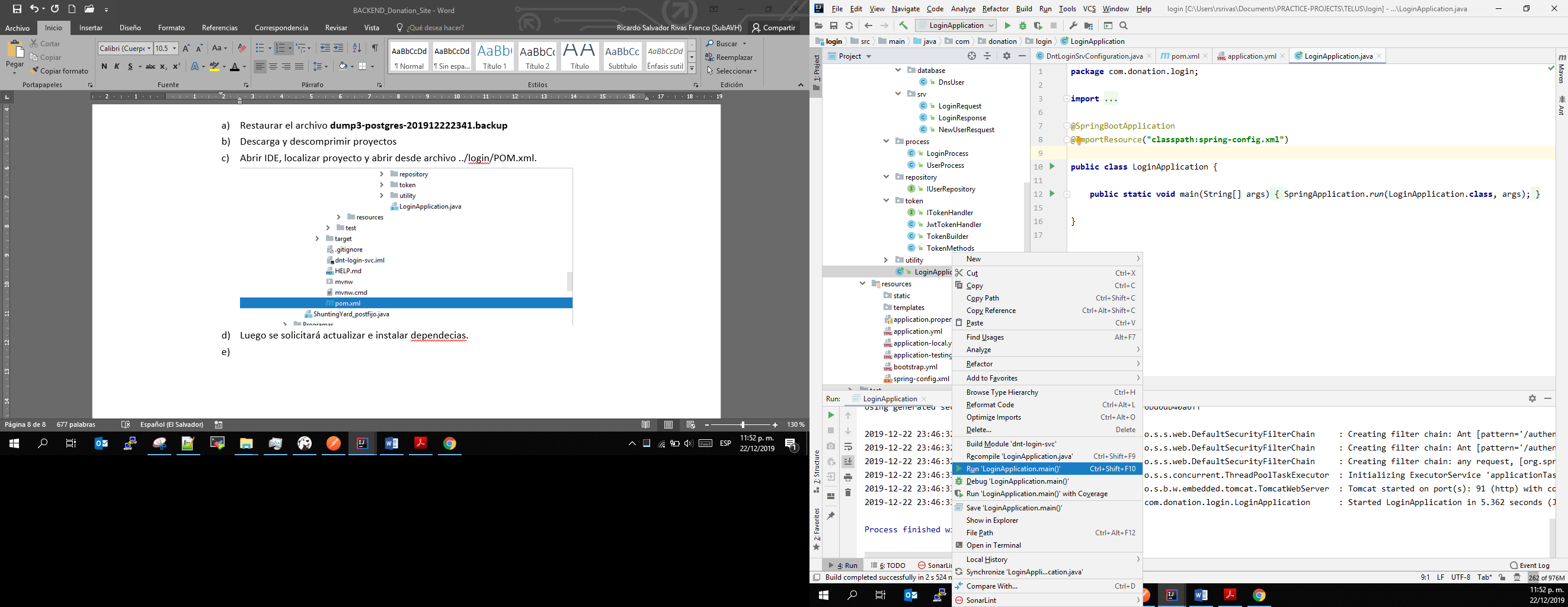
1. Luego se solicitará actualizar e instalar dependencias.
2. De ser necesario cambiar el usuario y contraseñas de archivo application-local.yml para la conexión a la base de datos.  
   
3. Por ultimo ubicarse en clase con el método main() y hacer clic en ejecutar.

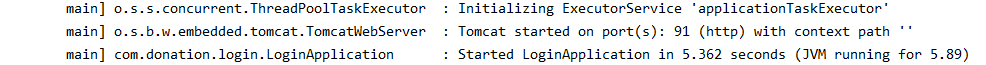
**dnt-integrator-svc:**

..\com\donation\dntintegratorsvc\DntIntegratorSvcApplication.java.

**dnt-login-svc:**

..\com\donation\login\LoginApplication.java



1. Luego aparecerá en consola el puerto en que se encuentra desplegado, esto se puede modificar en el archivo application.yml  
   
2. **Ubicación de archivo JAR:** Como otra alternativa se tiene el archivo JAR en la ubicación:  
   ..\dnt-integrator-svc\target\dnt-integrator-svc-0.0.1-SNAPSHOT  
   ..\login\target\login-0.0.1-SNAPSHOT.jar