**Proyecto final paso a paso**

1. Creé el repositorio “portfolio-Frontend” en GitHub.

**26/5/2022**

1. Creé la carpeta “portfolio-Frontend” en una carpeta de proyectos. La abrí en VS Code y armé un index.html básico.
2. Verifiqué la versión de Node.js (v16.14.2) con el comando “*node -v*” en la terminal de VS Code.
3. Instalé Typescript con el comando “npm install -g typescript”

**15/6/2022**

1. Abrí una terminal e hice un git init, un git add . y un git commit -m “Creado index.html”.
2. Comprobé que me encontraba en la rama “master” con “git branch”.
3. Cambié el nombre de la rama “master” a “main” con el comando “git branch -m main”.
4. Añadí el repositorio remoto con “*git remote add origin https://github.com/RAGiordano/portfolio-Frontend.git*”.
5. “git push -u origin main”.

**16/6/2022**

1. Instalé Bootstrap con “*npm install bootstrap*”.

**18/6/2022**

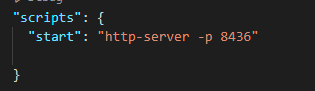
1. Cambié el título a “Portfolio Giordano Ricardo Agustín”.
2. Agregué la librería de Bootstrap al <head> con <link rel="stylesheet" href="bootstrap.min.css">
3. Creé la carpeta “IMG” y agregué las imágenes de logos para la barra de navegación.
4. Creé un contenedor para los elementos del portfolio.
5. Copié una Navbar de los componentes de Bootstrap.
6. Agregué los logos a la Navbar, ajusté los “height”, cambié los “alt”.
7. Agregué el banner, la foto y la información del header. Le di estilo en styles.css.

**19/6/2022**

1. Agregué los archivos con “git add .”, hice un commit -m “Header creado con bootstrap y css” y un “git push -u origin main”.
2. Controlé que estuviera instalado Node.js (comando “node”) y Typescript (comando “tsc -v”)
3. Para modificar el comportamiento predefinido del TSC ejecuté el comando **“tsc –init”**
4. **Para** crear una carpeta que contenga todos los archivos .js generados por el compilador TSC, descomenté la entrada “outFile” de tsconfig.json, le puse la ruta "./output/app.js" y comenté la entrada “module”.
5. Creé la carpeta “app” en el proyecto y el archivo “app.ts”. Después, escribí el código para mostrar por consola “Hello, World”, grabé y en la consola ejecuté el comando “tsc app/app.ts” para hacer la prueba de transpilar el código a un archivo app.js.
6. Creé un servidor de pruebas con el comando “npm install http-server”. Después lo inicié con “npm init”. En las preguntas, dejé las opciones por defecto con <ENTER> y en la confirmación puse “yes”. Esto me modificó el archivo package.json (si no lo hubiese tenido, lo habría creado).

**20/6/2022**

1. Configuré la entrada “scripts” como sigue:



A estas configuraciones las hago para hacer las pruebas del material de Argentina Programa y para incorporar las herramientas al proyecto.

1. Comenté la entrada “color-adjust: exact;” de bootstrap.css porque daba error.
2. Hice pruebas y no se ejecuta el código Javascript de output/app.js.

**06/7/2022**

1. Terminé de agregar las distintas secciones.
2. Incorporé los cambios a la rama principal:
   * git add .
   * git commit -m “maquetado inicial con bootstrap”

**07/7/2022**

1. “git push -u origin main”

**09/7/2022**

1. Instalación de Angular con “npm install -g npm@latest”
2. Instalación de Angular CLI con “***npm install -g @angular/cli@latest”***

**10/7/2022 ANGULAR**

1. Salí de VSCode, hice backup de la carpeta “portfolio-Frontend”, la renombré y creé otra vacía con el mismo nombre en la misma ubicación.
2. Abrí la nueva carpeta “portfolio-Frontend” con VSCode, abrí una terminal, me fui a la carpeta contenedora con “cd..” y después ejecuté el comando “***ng new portfolio-Frontend”.***
3. Cuando me preguntó si quería compartir datos de uso anónimos sobre este proyecto, puse que no (N). Cuando preguntó si quería agregar routing Angular, le puse que sí (y). En el formato de hoja de estilo seleccioné CSS.
4. Me dio **error** (“package install failed, see above”). Volví a ejecutar el comando. Esta vez me preguntó únicamente lo de routing y lo de CSS. Hice la misma selección y empezó la instalación. Otra vez dio error. Lo volví a hacer, y esta vez no dio error. El mensaje final fue “Successfully initialized git. Nothing to be done”.
5. Me moví a la carpeta del proyecto con “cd portfolio-Frontend” y después ejecuté la aplicación con “ng serve -o” (-o es lo mismo que –open). Me dio error:

Error [ERR\_MODULE\_NOT\_FOUND]: Cannot find module 'D:\PROYECTOS\ARGENTINA PROGRAMA\portfolio-Frontend\node\_modules\@angular\compiler\fesm2015\compiler.mjs' imported from D:\PROYECTOS\ARGENTINA PROGRAMA\portfolio-Frontend\node\_modules\@angular\compiler-cli\bundles\index.js

1. Probé varias soluciones, pero todas daban **error**. Opté por volver a cambiarle el nombre a la carpeta e iniciar un nuevo proyecto. Nuevamente abrir VSCode desde la carpeta contenedora, ejecutar el comando “ng new portfolio-Frontend”, seleccionar “Yes” y “CSS”, después moverme a la carpeta portfolio-Frontend y ejecutar “ng serve -o. Esta vez, no hubo ningún error y abrió la página por defecto de los proyectos Angular en el explorador (<http://localhost:4200/>).
2. Cerré todo. Volví a abrir VSCode en la carpeta del proyecto y me preguntó si quería instalar la extensión "Angular Language Service" para este repositorio. Le puse que sí.
3. Instalé Bootstrap con el comando “npm install bootstrap jquery @popperjs/core”.

Estos comandos **te permitirán tener las dependencias del Bootstrap, jQuery y el popperjs/core,** las cuales son necesarias para darle un mejor potencial al funcionamiento del framework.

Después seguí los siguientes pasos: en el archivo “angular.json” poner las siguientes instrucciones en los objetos “styles” y “scripts”:

* *"styles": [*
* *"node\_modules/bootstrap/dist/css/bootstrap.min.css",*
* *"src/styles.scss"*
* *],*
* *"scripts": [*
* *"node\_modules/jquery/dist/jquery.min.js",*
* *"node\_modules/@popperjs/core/dist/umd/popper.min.js",*
* *"node\_modules/bootstrap/dist/js/bootstrap.min.js"*
* *]*

**Hasta ahora, todo Ok.**

1. Ejecuté los pasos (5) al (9) para inicializar Git y actualizar mi proyecto en GitHub. Me dio **error** en el “git push” por los archivos existentes en el repositorio remoto. Entré a GitHub, renombré el repositorio y generé otro vacío con el mismo nombre. Entonces me dejó.

**11/7/2022**

1. Probé de cargar los archivos de GitHub en Stackblitz usando la dirección <https://stackblitz.com/github/RAGiordano/portfolio-Frontend>

**13/7/2022**

1. Cambié el título de la página a “Portfolio Giordano Ricardo Agustín”.

**14/7/2022**

1. Generé el componente “header” y una carpeta “components” para ubicar todos los componentes de acá en adelante. Comando: ng g c components/header.

Se genera la carpeta y los archivos.

**17/7/2022**

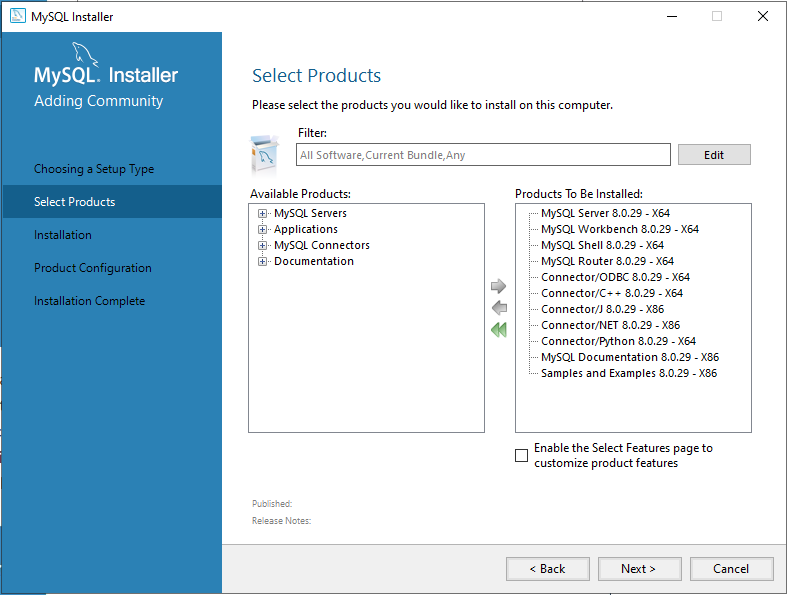
1. Descubrí que Angular no toma las imágenes si no están en ciertas carpetas. Creé la carpeta “Logos” dentro de “assets”. Copié los logos en esa carpeta y otras imágenes dentro de “assets”.
2. Empecé a mover el código que había generado hasta el 7/7/2022. Copié lo del header (barra de navegación, en realidad) dentro del HTML. Copié la parte general del CSS en “styles.css” y la parte propia del header en “header.component.css”. Agregué comentarios y separé los estilos por categorías (colores, formatos de texto, separaciones, etc.). La barra de navegación funciona bien.
3. Generé el componente “personal” con “ng g c components/personal”.
4. Copié el código del componente y lo propio del estilo.
5. Controlé que funcionara todo bien e hice un “git add .” y un “git commit -m “versión inicial de maquetado estático terminada””.
6. Hice un “git push -u origin main”.
7. Probé que funcionara <https://stackblitz.com/github/RAGiordano/portfolio-Frontend>. Todo ok.

**23/7/2022**

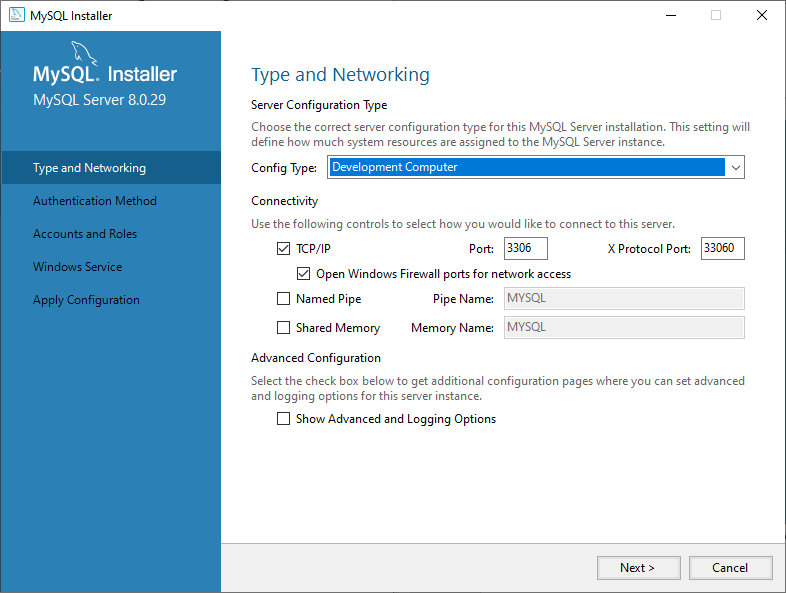
1. Descargué e instalé MySQL Workbench
2. Hice el DER en MySQL Workbench
3. Descargué e instalé Xampp (servidor Apache)

**25/7/2022**

1. Agregué la línea “skip-grant-tables” debajo de “[mysqld]” dentro del archivo de configuración de Xampp, cuya ruta es “c:\xampp\mysql\bin\my.ini”.
2. Le di stop al servidor sql de xampp, eliminé la conexión de MySQL Workbench y la volví a crear. Omití la advertencia por la versión.
3. Como MySQL Workbench no me abre la base de datos y no tengo el servicio de MySQL, desinstalé MySQL y lo volví a instalar con las siguientes opciones:



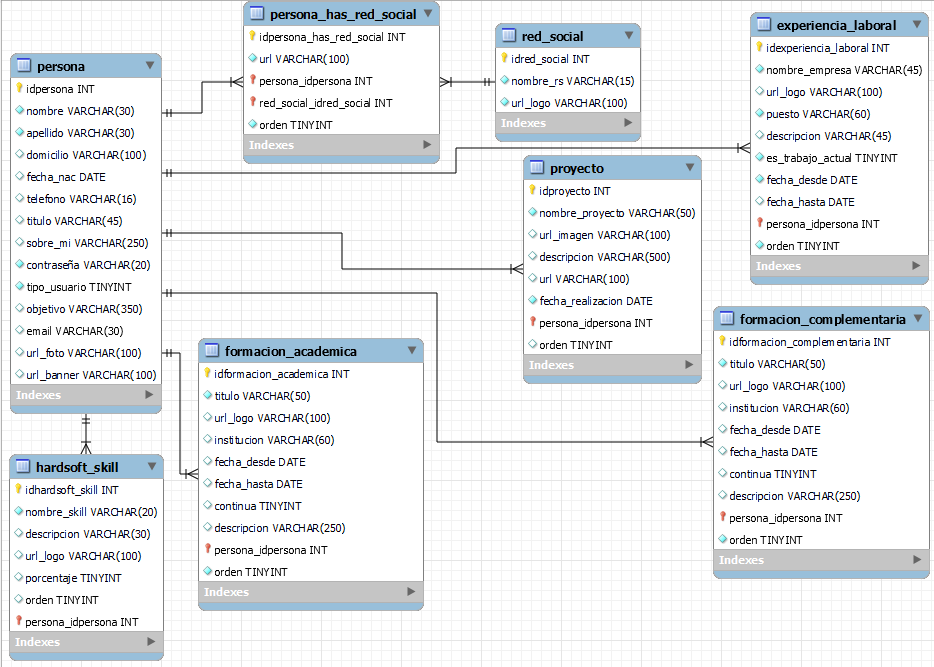
1. Al terminar la instalación, me pidió configuraciones:



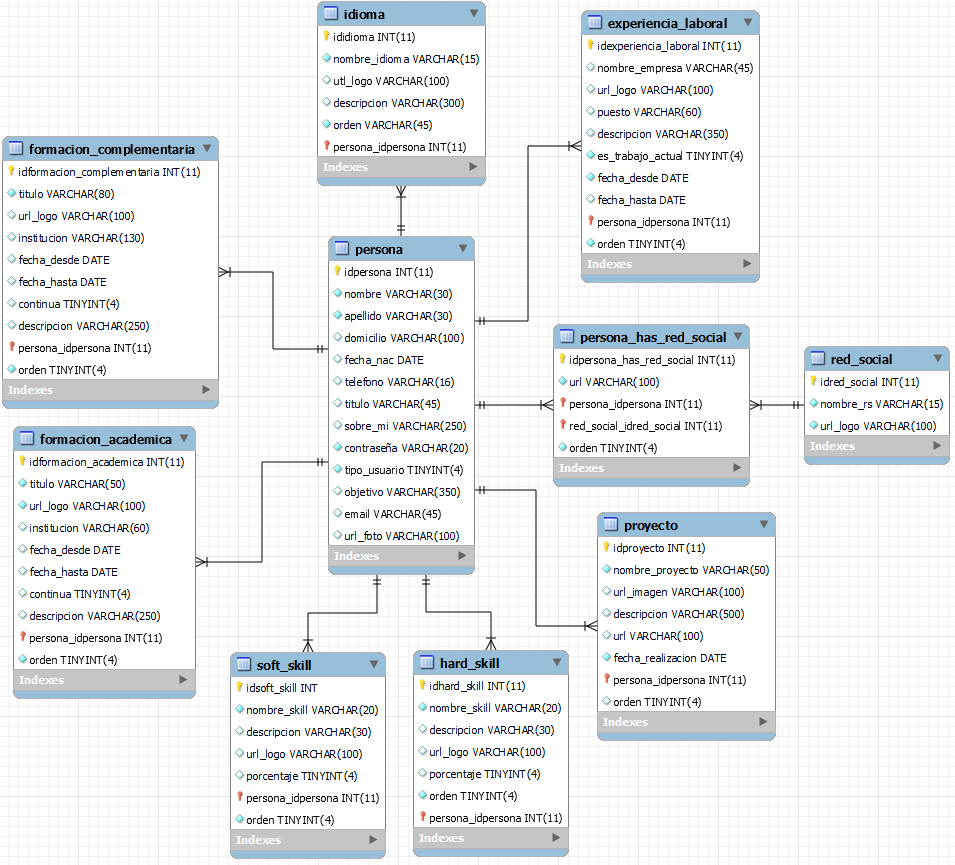
1. Dejé las configuraciones por defecto. La contraseña de root es RAGSQL
2. Me di cuenta de que me había faltado el paso de ir al menú Database - Forward Engineer. Ahí avancé con las opciones por defecto. Finalizó sin errores. Un error común en este paso es en la creación de las claves foráneas. En ese caso, hay que volver atrás y borrar donde dice “INDEX [clave foránea] … VISIBLE”.
3. Después de eso, fui al menú Database – Connect to database…” y me conecté.
4. Después fui a la pestaña de la conexión local. Allí, en la pestaña “Schemas” de la barra izquierda, aparece la base de datos.
5. Generé una consulta para cargar información en “persona”.

**26/7/2022**

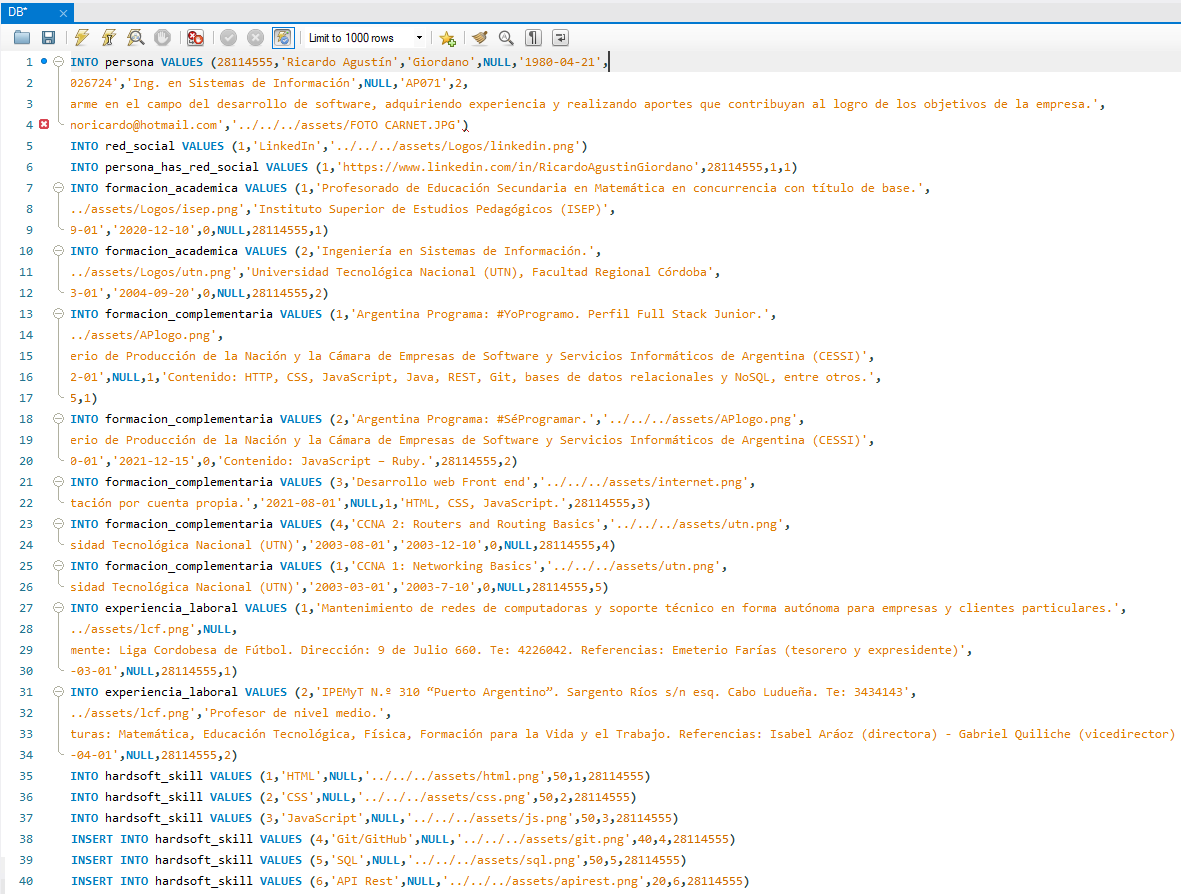
1. Agregué campos que faltaban a la base de datos.



1. Empecé a cargar datos y me di cuenta que necesitaba cambiar algunas tablas y campos. El DER quedó así:



1. El código para la primera carga de datos quedó así:



**1/8/2022**

1. Creé una cuenta en Oracle
2. Descargué e instalé Java Runtime Environment 8u341.
3. Descargué e instalé Java Development Kit 8u341.
4. Descargué e instalé Apache Netbeans LTS 12.0

**4/8/2022**

1. Siguiendo las instrucciones del curso de Java de Lucas Moy:
   * Copié la ruta de instalación de JDK (C:\Program Files\Java\jdk1.8.0\_341\bin)
   * Clic derecho en Mi Pc 🡪 Más… 🡪 Propiedades 🡪 Configuración avanzada del sistema 🡪 Variables de entorno.
   * Dentro de las variables de sistema, puse “Nueva…”. 🡪 Nombre: JAVA\_HOME 🡪 Valor: C:\Program Files\Java\jdk1.8.0\_341\bin 🡪 Aceptar.
   * También dentro de las variables de sistema, busqué la variable “Path” 🡪 la selecciono 🡪 clic en “Editar…” 🡪 “Nuevo” 🡪 Pegar nuevamente la ruta C:\Program Files\Java\jdk1.8.0\_341\bin 🡪 Aceptar.
   * Dentro de las variables de sistema, puse “Nueva…”. 🡪 Nombre: JAVA\_HOME 🡪 Valor: C:\Program Files\Java\jdk1.8.0\_341 🡪 Aceptar. Este paso es muy parecido al anterior del anterior, pero sin el “/bin” al final de la ruta.
   * Para probar si está todo bien, abro una ventana de comandos CMD y pongo “javac -version”. Si está todo bien, me sale la versión de JDK instalada.
2. Abrí Netbeans e hice una prueba para ver si funcionaba el “Hola, mundo”. No funcionaba.
3. Tuve problemas con NetBeans. Lo desinstalé, al igual que a JDK. Reinstalé JDK (misma versión) y después instalé NetBeans 12.5.
4. Volví a probar mi “Hola, mundo” y esta vez funcionó.

**19/8/2022**

1. Cambié los colores de NetBeans instalando los colores de Sublime.
2. Empecé el ejercicio del video de la clase 7, para lo cual configuré el Xampp en el puerto 3016.

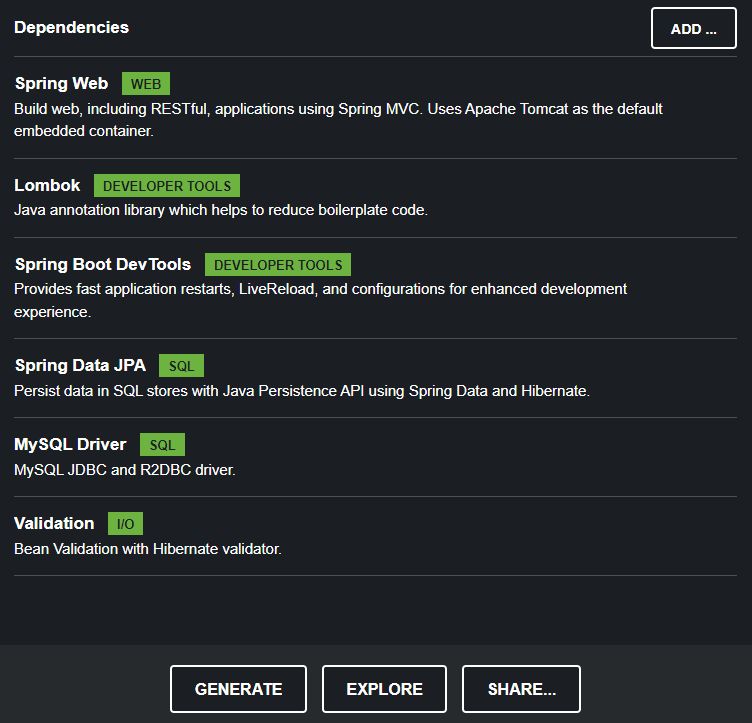
**21/8/2022**

1. Cambié algunos colores de NetBeans que eran molestos:
   * Fonts&Collors --> Syntax -->
   * JSP Scriptled Delimiter (Green/Inherited)
   * Mark Occurrences (Black/[255,255,204])
   * Error (White/[255,153,153])
2. Cuando en un request me devuelve un objeto y yo necesito que tenga formato de lista, hago un “casteo”, agregando *(List)* antes del request. En otro caso, pasó lo mismo, pero con un string, cuando se esperaba un entero. Ahí se solucionó con un Integer.parseInt(expresión).
3. Cuando quise correr el proyecto en un momento, apareció la advertencia del antivirus AVG, que mandó el archivo catalina.bat.Y a cuarentena (de una carpeta temporal). Para solucionarlo, fui a Xampp, detuve el módulo Apache y lo volví a iniciar.

**22/8/2022 – CLASE 8 – SPRING BOOT**

1. SpringBoot:
   * fui a <https://start.spring.io/>
   * Maven Proyect
   * Java
   * Versión de Spring Boot 2.7.3
   * Metadata:
     + Group: com.portfolio
     + Artifact: RAG
     + Name: Ricardo A. Giordano (si pongo Agustín con tilde no genera el proyecto)
     + Description: Portfolio Web de Ricardo Agustín Giordano
     + Package name: com.portfolio.RAG
     + Packaging: Jar
     + Java: 8
     + Dependencias:
       1. **Spring Web** (para crear aplicaciones Web, incluye RESTful y usa Spring MVC y Apache Tomcat)
       2. **Spring Data JPA** (persistencia de datos en almacenes SQL con la API de persistencia de Java mediante Spring Data e Hibernate)
       3. **MySQL Driver** (para conectar la base de datos con el proyecto)
       4. Validation (validación de Beans con el validador de Hibernate)
       5. **DevTools** (herramientas de desarrollador; proporciona reinicios rápidos de aplicaciones, LiveReload y configuraciones para mejorar la experiencia de desarrollo)
       6. **Lombock** (biblioteca de anotaciones de Java que ayuda a reducir el código repetitivo; por ejemplo, con @Getter @Setter nos genera todos los getters y setters automáticamente)
       7. **H2 Database** (base de datos local para algunas pruebas)





1. En Other Sources 🡪 src/main/resources 🡪 <default package> 🡪 application.properties

*spring.jpa.hibernate.ddl-auto=update*

*spring.datasource.url=jdbc:mysql://localhost:3306/portfolioap?useSSL=false&serverTimezone=UTC*

*spring.datasource.username=root*

*spring.datasource.password=*

*spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect*

1. En Workbench creé una base de datos vacía (con el nombre que indiqué en NetBeans (“portfolioap”).
2. En NetBeans, dentro de *com.portfolio.RAG* creé los siguientes paquetes Java:
   * *com.portfolio.RAG.Entity*
   * *com.portfolio.RAG.Repository*
   * *com.portfolio.RAG.Interface*
   * *com.portfolio.RAG.Service*
   * *com.portfolio.RAG.Controller*

**25/8/2022**

1. Creé la clase *Persona* dentro de *Entity*.
2. Definí los atributos de persona.
3. Usé las anotations *@Getter* y *@Setter* de lombok para no tener que generar los getters y los setters.
4. Generé la interfaz *IPersonaRepository* en *Repository*, que extiende de *JpaRepository*.
5. Generé la interfaz *IPersonaService* en *Interface*.
6. Definí los métodos para traer lista de personas, borrar, buscar y agregar y modificar.
7. Generé la clase *ImpPersonaService* en *Service*.
8. Definí los métodos para pasar lista de personas, agregar o modificar persona, borrar y encontrar persona por id.
9. Generé la clase *PersonaController* dentro de *Controller*.
10. Definí las funciones para traer una lista de personas, crear persona, borrar persona y modificar persona

**REPLICAR LO DE PERSONAS PARA TODAS LAS OTRAS TABLAS**

**26/8/2022**

1. En el frontend importé *HttpClientModule* en *app.module.ts*:

import { HttpClientModule } from '@angular/common/http'

También *HttpClientModule* en “Imports”.

1. Creé la carpeta “model” en “app” para alojar las clases de los objetos.
2. Creé el archivo “persona.model.ts”, defino en él a la clase *persona* y la exporto.

**27/8/2022**

1. Cargué los datos de una persona con Postman (<https://web.postman.co/>), mediante una petición “CREAR PERSONA”, con los siguientes datos (JSON):

{"dni":"28114555",

"nombre":"Ricardo Agustin",

"apellido":"Giordano",

"domicilio":"Bartolome Piceda 991. Chacra Escondida.",

"fecha\_nac":"21/04/1980",

"telefono":"3513026724",

"presentacion":"Desarrollador Full Stack",

"sobre\_mi":"Sobre mi",

"contrasena":"contrasena",

"tipo\_usuario":"tipo usuario",

"objetivo":"objetivo",

"email":"email",

"url\_foto":"url\_foto",

"url\_banner":"url\_banner"}

**NetBeans**:

1. Definí el método findPersona en PersonaController, que trae los datos de la primera persona, para hacer la prueba de conectar el frontend y traer nombre y apellido desde la base de datos.

**VSCode**:

1. Generé la carpeta “services” y el servicio “portfolio.service.ts” dentro de ella.
2. En “portfolio.service.ts” definí la clase *PortfolioService*. Declaré la variable *url* y la inicialicé con la dirección de la página. Definí el método *getPersona*.
3. Problema de CORS. Solucionado con *@CrossOrigin(origins = "http://localhost:4200")* en “PersonaController.java” (NetBeans).

Prueba exitosa.

**01/9/2022**

1. Renombré el componente “personal” a “encabezado-personal” para que sea más representativo.

**02/9/2022**

1. Agregué la librería de Font Awesome para usar los íconos de edición.
2. Generé los componentes:

* acerca-de
* experiencia
* formacion-academica
* formacion-complementaria
* habilidades
* proyectos
* footer

Estos componentes se suman a los ya existentes:

* header
* encabezado-personal

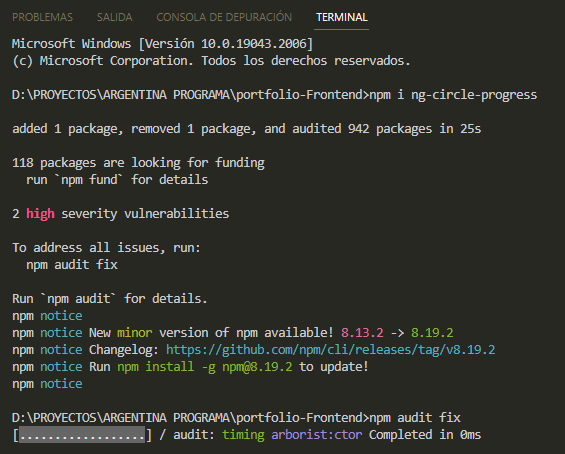
**03/9/2022**

1. Moví el código de cada componente a su lugar correspondiente.
2. Generé los módulos “home” y “login”.

**19/9/2022 al 21/9/2022**

1. Terminé la primera versión del maquetado y diseño responsivo del header y las secciones objetivo-acerca de, experiencia y capacitación.
2. Avancé con la sección de hard & soft skills.
3. Importé ng circle progress con el comando: npm i ng-circle-progress

Me dio el siguiente error:



Trato de solucionarlo con el comando npm audit fix pero no funciona.

Pruebo una copia anterior y tiene el mismo error.

**22/9/2022**

1. Problema de ayer solucionado con *npm update*.
2. *npm audit fix*
3. *npm install ng-circle-progress --save*

Once you have installed it, you can import it in any Angular application,

then from your Angular **AppModule**:

// Import ng-circle-progress

import { NgCircleProgressModule } from 'ng-circle-progress';

// Specify ng-circle-progress as an import

NgCircleProgressModule.forRoot({

// set defaults here

radius: 100,

outerStrokeWidth: 16,

innerStrokeWidth: 8,

outerStrokeColor: "#78C000",

innerStrokeColor: "#C7E596",

animationDuration: 300,

...

})

**24/9/2022**

1. Maquetado de habilidades con circle progress.

**25/9/2022**

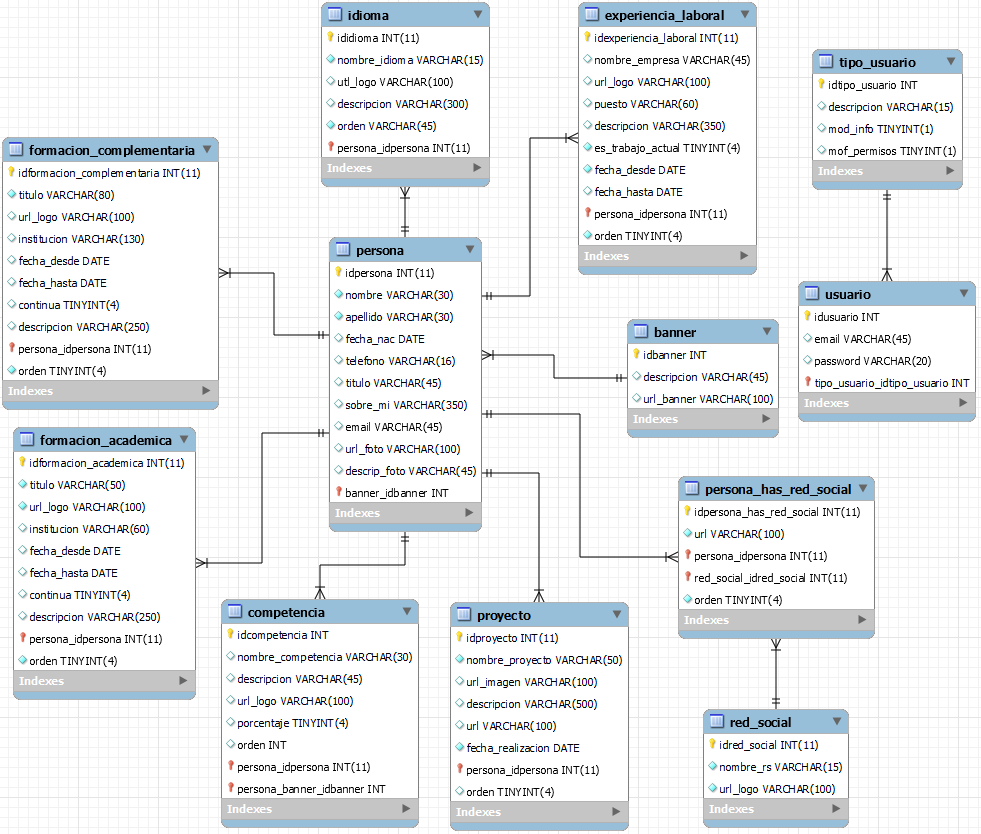
1. Maquetado de proyectos y footer.
2. Controlé que funcionara todo bien e hice un “git add .” y un “git commit -m “versión mejorada de maquetado estático y encabezado personal conectado a BD””.
3. Hice un “git push -u origin main”.

**30/9/2022**

1. Ajustes en base de datos (agregué descrip\_foto y quité dirección en persona).

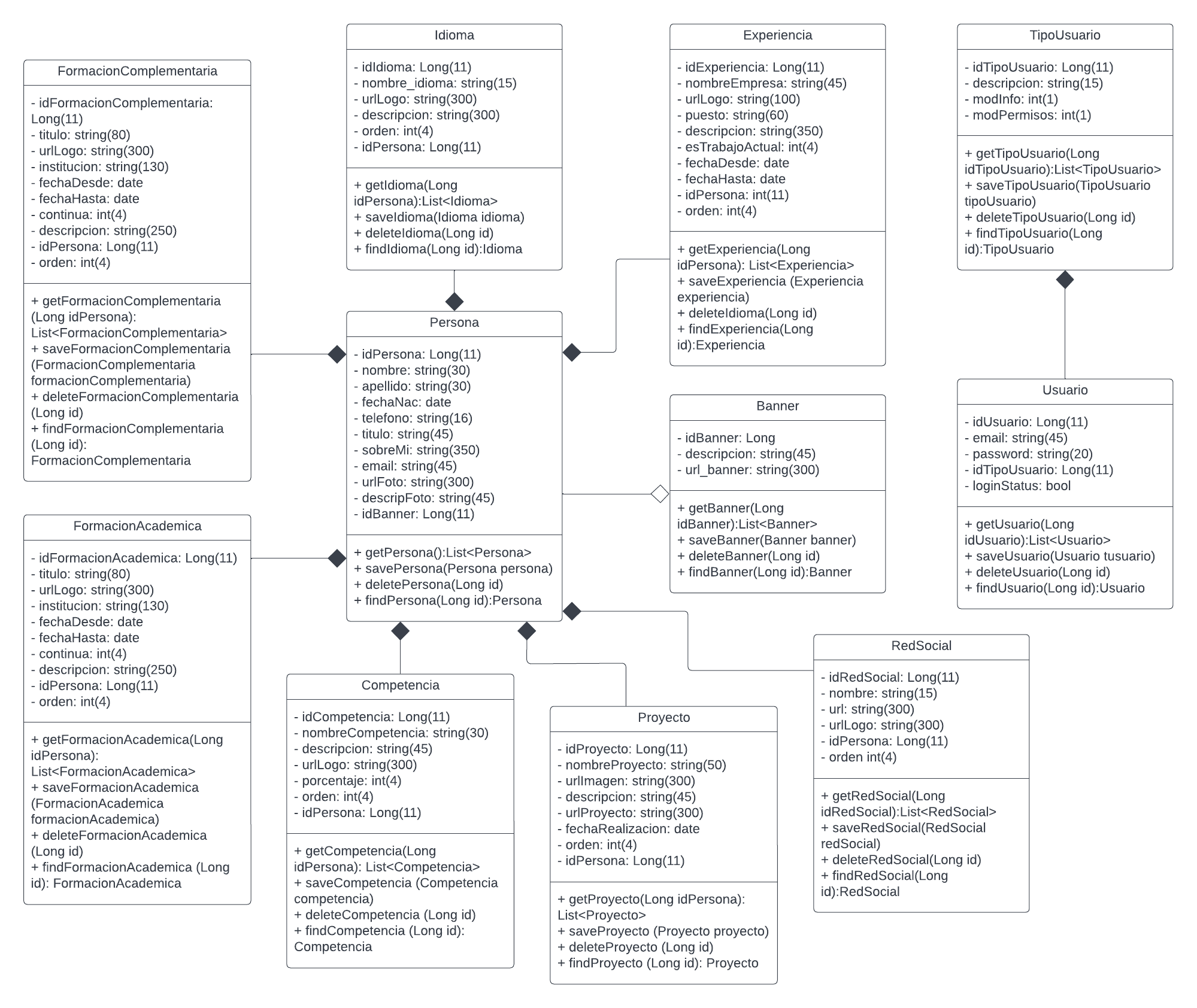
**13/10/2022**

1. DER actualizado:



**19/10/2022**

1. Terminé el modelo UML:



<https://lucid.app/lucidchart/e8d73cec-67ab-492c-9c96-17a325ed7c0d/edit?viewport_loc=-366%2C-8%2C2675%2C1217%2C0_0&invitationId=inv_f1d93e3e-f282-477e-91b3-314dd605c298>

1. Subí el diagrama de clases a GitHub con el commit “Diagrama de Clases UML” y agregué el enlace a Lucid Chart en los comentarios.

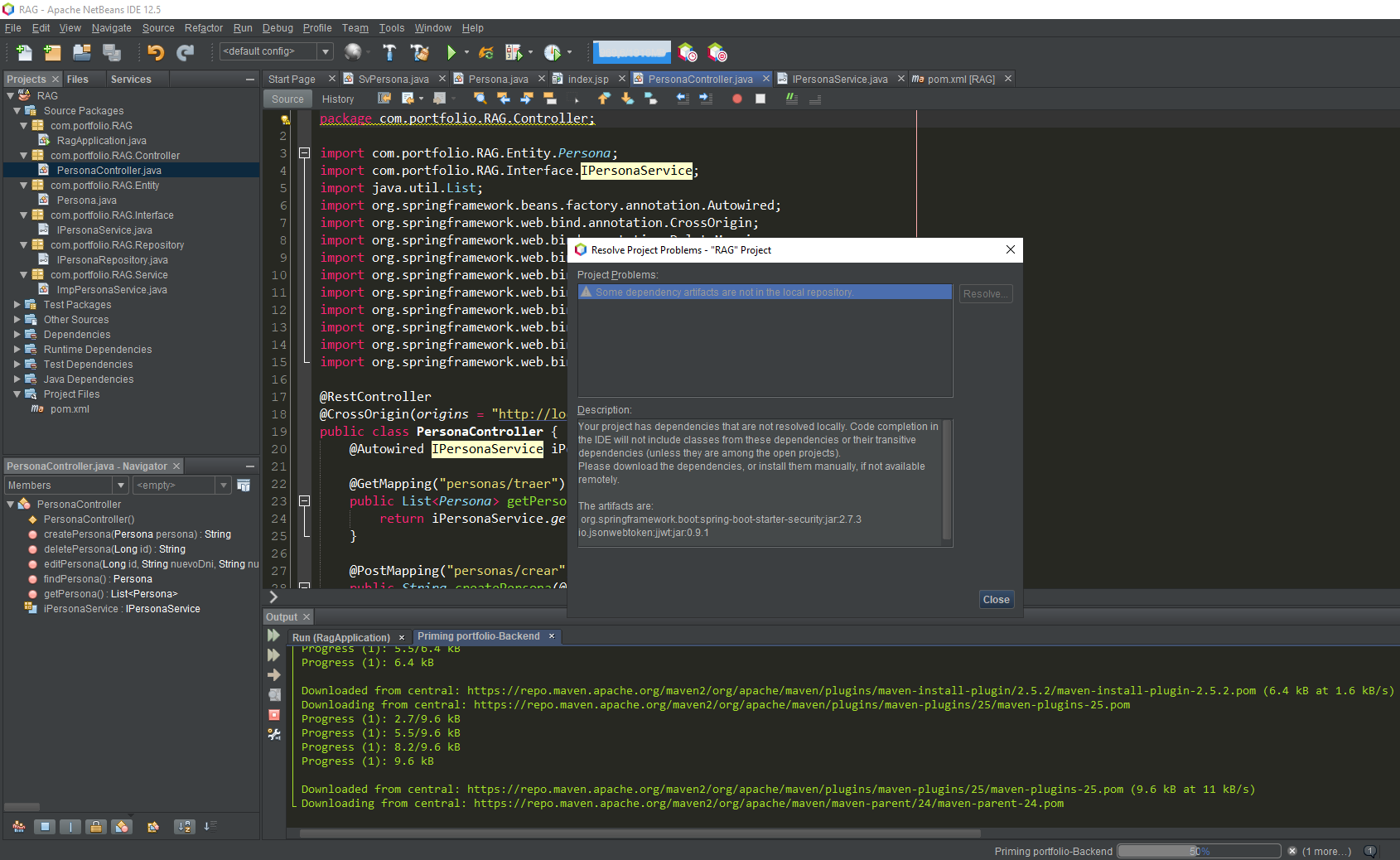
**22/10/2022**

1. Generé el componente iniciar-sesion y el componente portfolio.
2. Copié en cada uno de los templates de los nuevos componentes la ruta de los módulos respectivos.
3. Agregué las rutas en app-routing.module.ts y configuré la ruta por defecto.
4. Importé ReactiveFormsModule en app.module.ts.
5. En iniciar-sesion.component.ts hice la importación: import {FormBuilder, FormGroup, Validators} from '@angular/forms';
6. Copié un formulario de login de la página de Bootstrap en iniciar-sesion.component.html.
7. Agregué las validaciones en el formulario (campos requeridos, formato de email y contraseña de 8 caracteres o más, etc.). Me aparecen errores por el \*ngIf:

Parser Error: Conditional expression Email?.errors && Email?touched requires all 3 expressions at column 46 in [{'is-invalid': Email?.errors && Email?touched, 'is-valid': !Email?.errors && Email?.touched}] in d:/PROYECTOS/ARGENTINA PROGRAMA/portfolio-Frontend/src/app/iniciar-sesion/iniciar-sesion.component.html@5:23

**23/10/2022**

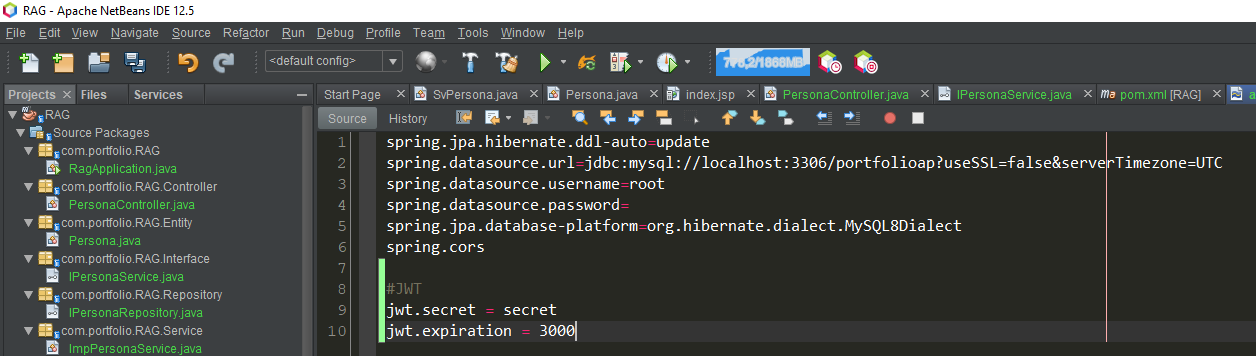
1. Inyecté la dependencia “spring-boot-starter-security” versión 2.7.5 en forma manual al pom.xml. Copié el texto desde <https://mvnrepository.com>
2. Inyecté la dependencia “JSON Web Token Support For The JVM” versión 0.9.1.
3. Me aparece una advertencia sobre las dependencias que no están en el repositorio local. Pongo que lo solucione (desde la lamparita).



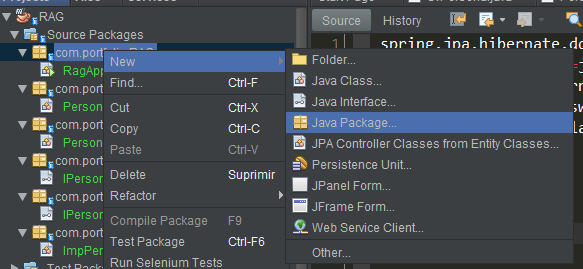
1. En GitHub, renombré la carpeta de backend y creé una nueva, para mover todo desde cero.
2. Abrí una ventana con el símbolo de sistema (CMD).
3. Comprobé que me encontraba en la rama “master” con “git branch”.
4. Cambié el nombre de la rama “master” a “main” con el comando “git branch -m main”.
5. Ejecuté los comandos:

* git init
* git add .
* git commit -m "Backend con controlador, modelo, repositorio y servicio de Persona funcionando"
* git remote add origin https://github.com/RAGiordano/portfolio-Backend.git
* git push -u origin main

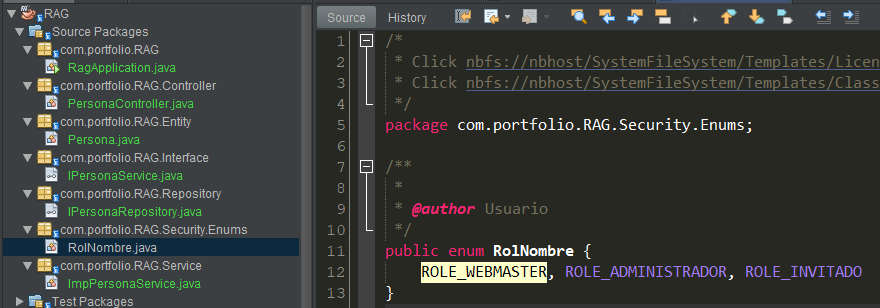
1. Actualicé algunos campos de la base de datos. Generé los distintos modelos (DER, modelo de clases) y lo subí a GitHub con el commit "Modelos de base de datos".
2. Comienzo a configurar el JSON Web Token (JWT), primero en el archivo application.properties:



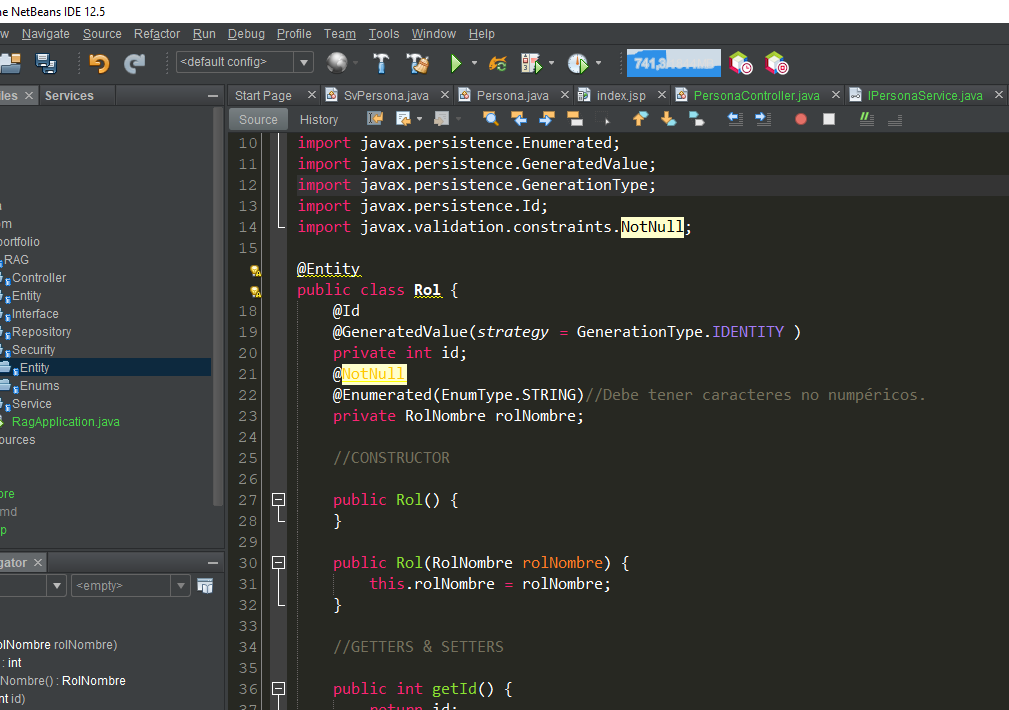
1. Creé el paquete Security dentro de com.portfolio.RAG



1. Sobre Security, creamos un nuevo paquete “Enums”.
2. Dentro del paquete Enums, generamos una nueva Java.Class “RolNombre”.
3. Editamos RolNombre:



1. En la parte de archivos de NetBeans, en Security, generé un paquete Enums, y dentro de este una Java Class “Rol.Java”.
2. Editamos la clase Rol, agregando atributos, constructores, getters y setters:



1. Generé otra clase Java dentro del paquete Entity, llamada “Usuario”.

**25/10/2022**

1. Usé las explicaciones de mgbcomputación para hacer la parte de login con JWT en el backend con NetBeans.

Me aparece tachada la librería WebSecurityConfigurerAdapter. Investigué que es por la versión, pero funciona igual. No levanta el Tomcat.

**27/10/2022**

1. No levantaba el Tomcat. El error que figuraba era “Error creating bean with name 'jwtTokenFilter'”. Lo solucioné corrigiendo el archivo JwtProvider.java, donde había puesto unos paréntesis que en realidad tenían que ser llaves. Tiene que quedar así:

@Value("${jwt.secret}")

private String secret;

@Value("${jwt.expiration}")

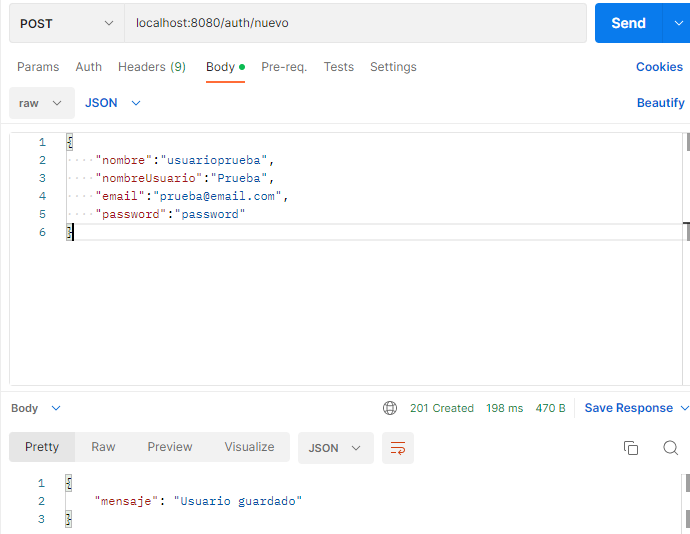
private int expiration

También hice algunos cambios sugeridos por otros compañeros en el archivo MainSecurity.java:

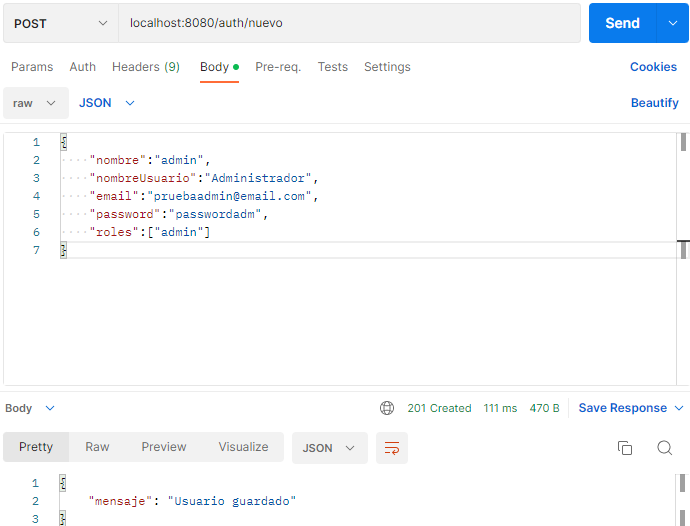
* Sacar el espacio que había después de la coma en replace("Bearer","")
* Cambiar el path en antMatchers por "/\*\*"

1. Hice pruebas con Postman.

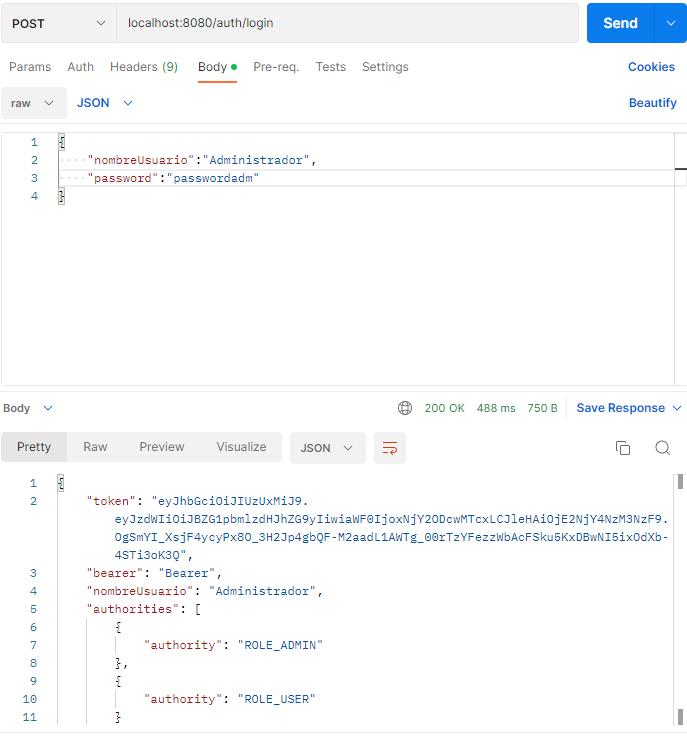
Creé el usuario:



Creé el administrador (agrego “roles” porque “user” se pone por defecto, no así “admin”):



Probé el login con el rol de administrador:



Hago lo mismo con el otro usuario y todo OK

Hago pruebas con usuarios inexistentes y no me permite loguearme. Todo OK.

1. Dentro de PersonaController.java, voy a restringir la creación, eliminación y modificación para los usuarios con rol ADMIN

@PreAuthorize("hasRole('ADMIN')")

1. Hago pruebas en Postman para crear y borrar personas estando logueado como administrador y como usuario y todo funciona OK.
2. Genero la clase NuevoUsuario

PS D:\PROYECTOS\ARGENTINA PROGRAMA\portfolio-Frontend\src\app\model> ng g class nuevo-usuario --skip-tests

1. Declaro los atributos de la clase que acabo de crear.
2. Genero la clase LoginUsuario (de manera similar a NuevoUsuario).
3. Declaro los atributos y hago un constructor.
4. Genero la clase JdtDto con sus atributos (también en la carpeta model).
5. En la carpeta Services genero el servicio token y lo configuro.
6. Genero y configuro el servicio auth.
7. Configuro el componente de login.
8. Configuro los botones y el ts de header.

**28/10/2022**

1. Generé el servicio InterceptorService en services.
2. Agregué interceptorProvider en “providers” en app.modules.ts

**29/10/2022**

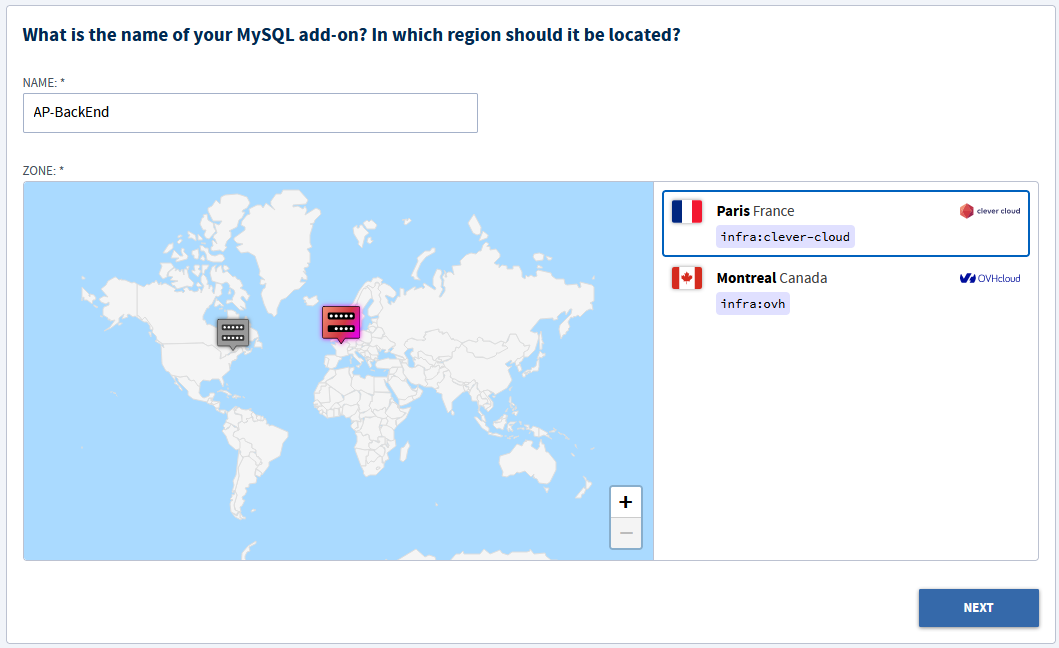
1. En NetBeans generé la clase Experiencia en Entity.
2. Generé la interface RExperiencia en Repository.
3. Generé la clase SExperiencia en Service.
4. Creé el paquete Dto
5. Dentro de Dto, generé la clase DtoExperiencia
6. Agregué manualmente la dependencia commons-lang3 en el pom.
7. Generé la clase CExperiencia en Controller.
8. Edité el servicio portfolioService para agregar lo de experiencia laboral
9. Edité el ts y el html del componente experiencia
10. Creé el componente nueva-experiencia dentro de la carpeta experiencia
11. Configuré el ts de nueva-experiencia y hago el formulario para agregar experiencia en el html de nueva-experiencia.
12. Agregué la ruta para el formulario de nueva experiencia en app-routing.module.ts.
13. Le di formato básico al formulario de nueva experiencia.
14. Instalé la dependencia de íconos de Bootstrap con *npm i bootstrap-icons*
15. Creé el componente para editar experiencia laboral
16. Configuré el ts de editar-experiencia.
17. Copié y adapté el formulario que tenía para nueva experiencia, para usarlo en editar.

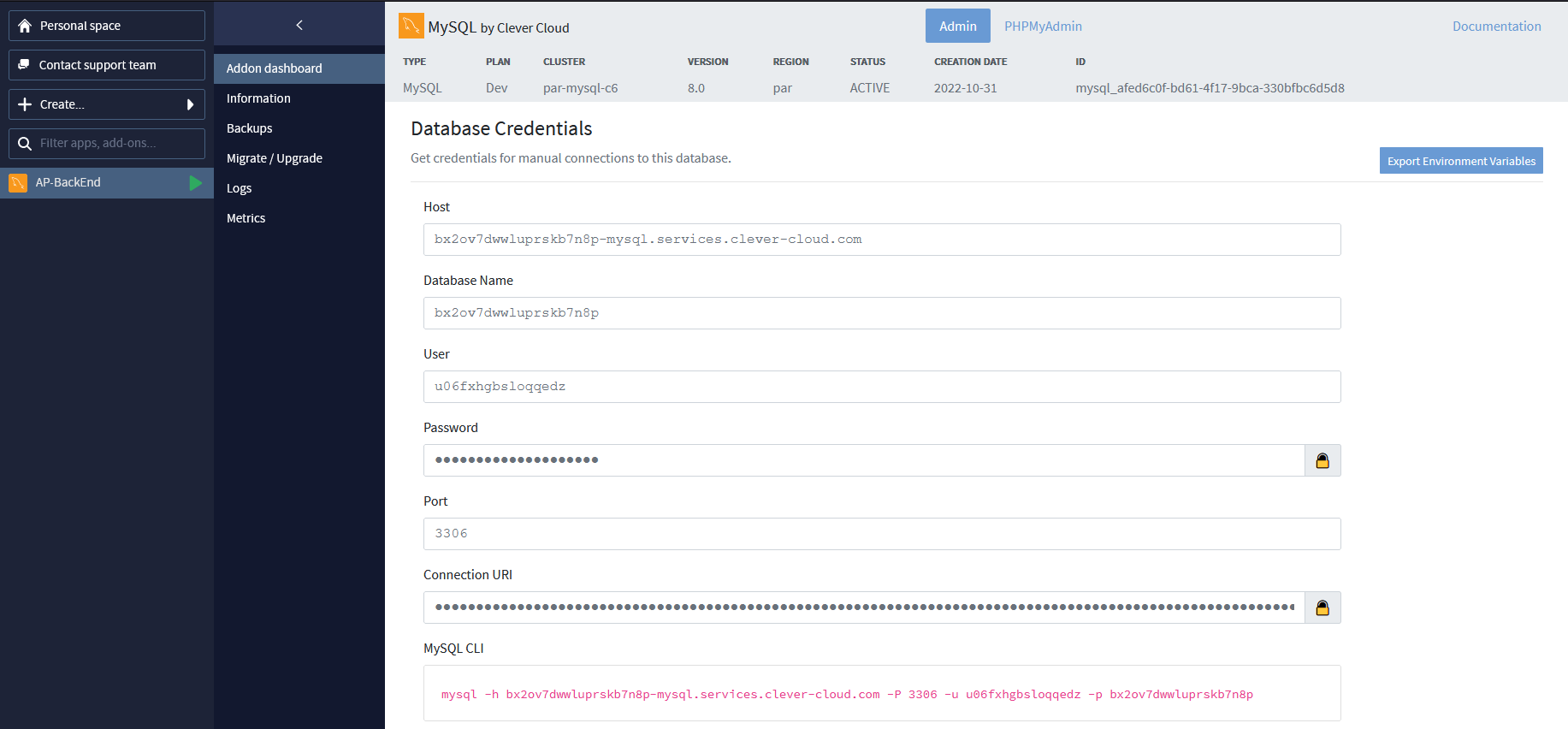
**30/10/2022**

1. Codifiqué el backend y el frontend de la sección Formación Académica de manera similar a Experiencia Laboral.
2. Codifiqué el backend y el frontend de la sección Competencias. Todo OK.

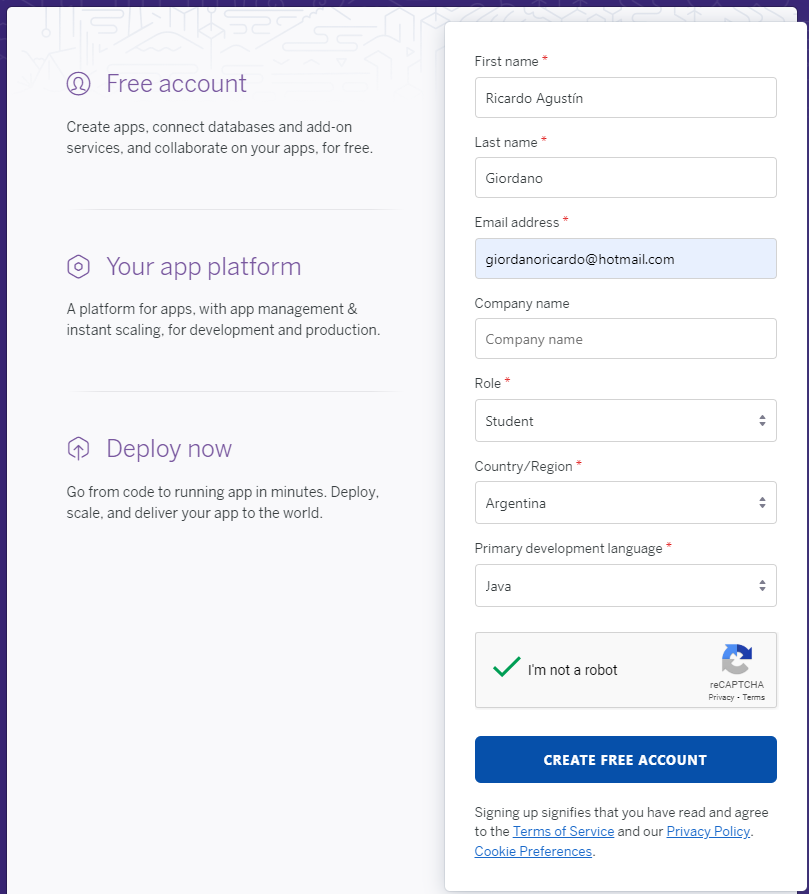
**31/10/2022**

1. Modifiqué todo lo relacionado a Persona en el backend y en el frontend.
2. Hice las adecuaciones en el encabezado personal para poder modificar.
3. Codifiqué el backend y el frontend de la sección Proyectos. Todo OK.
4. Creé una cuenta en CleverCloud.
5. Generé un nuevo add-on

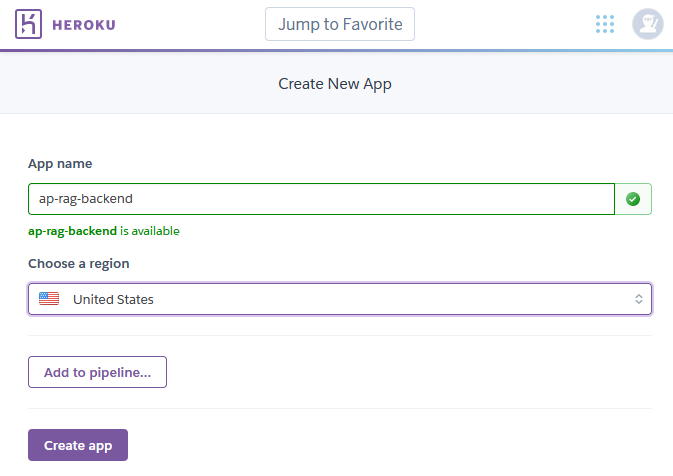




1. Hice las configuraciones correspondientes en NetBeans con la información aportada por CleverCloud.
2. Corrí la aplicación en NetBeans para que cree las bases de datos.
3. Me registré en Heroku



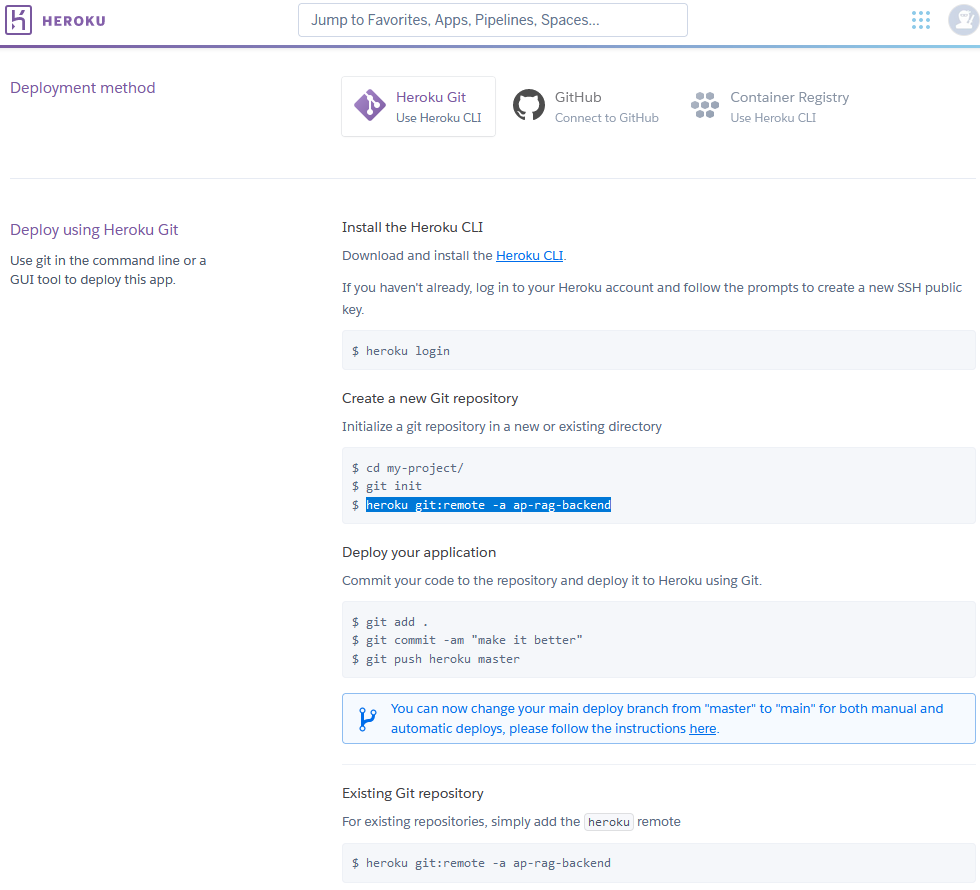
1. Creé una nueva App en Heroku

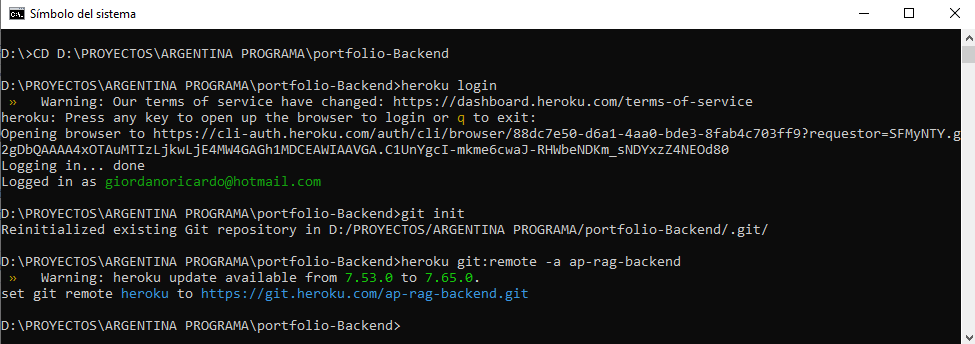


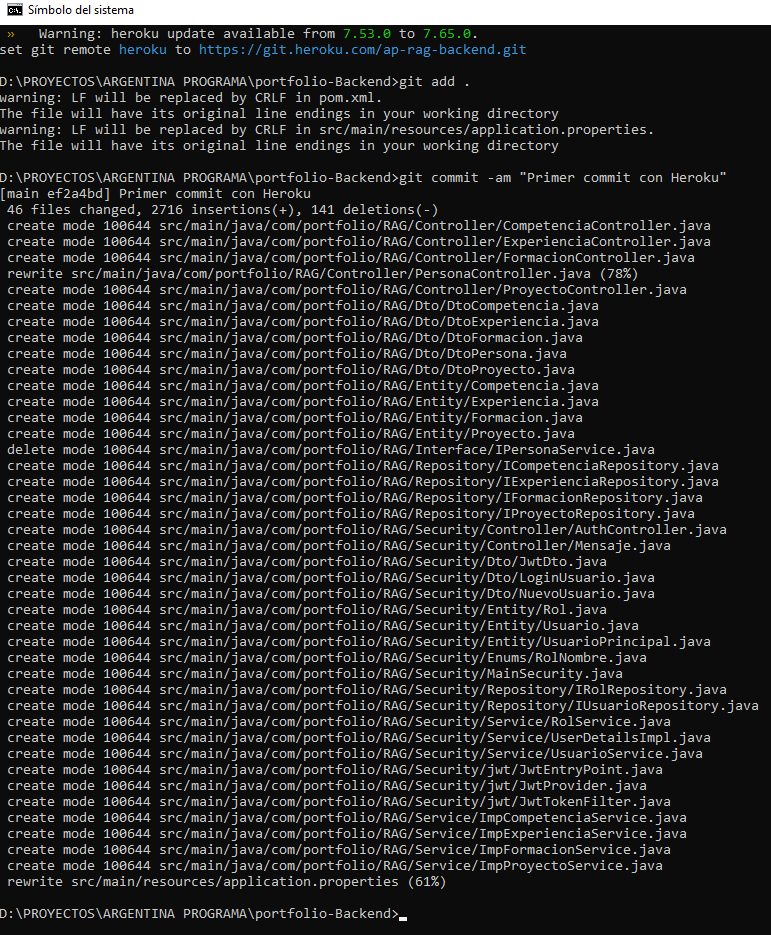
1. Instalé Heroku Cli desde:

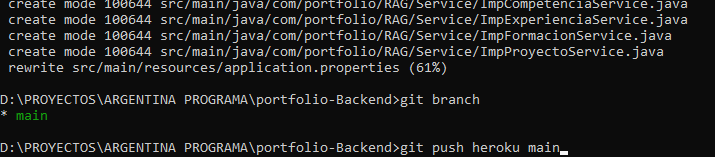
<https://devcenter.heroku.com/articles/heroku-cli#download-and-install>

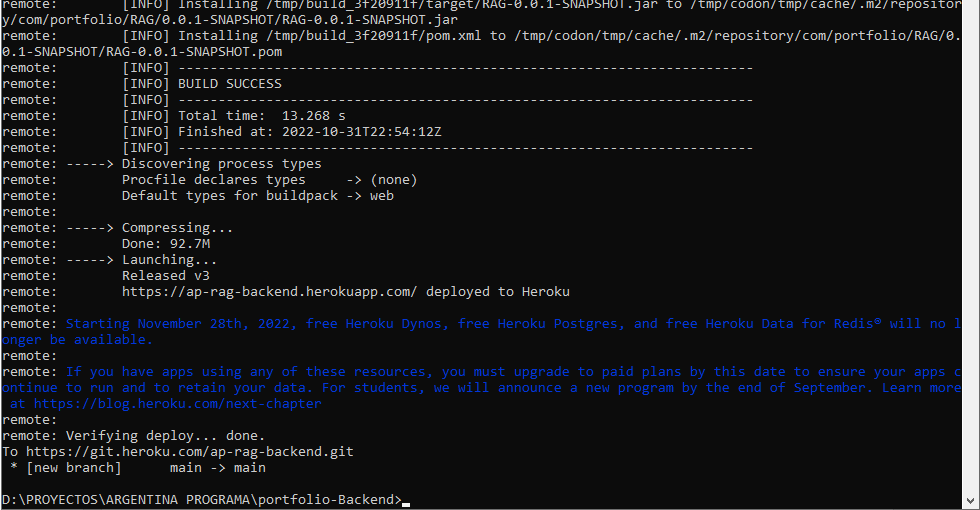
1. Veo y sigo los pasos que indica Heroku







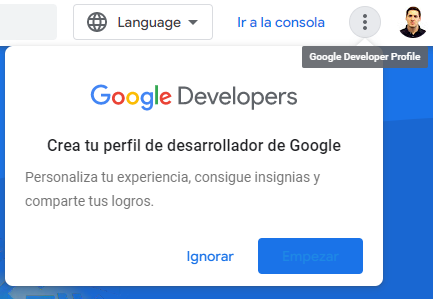


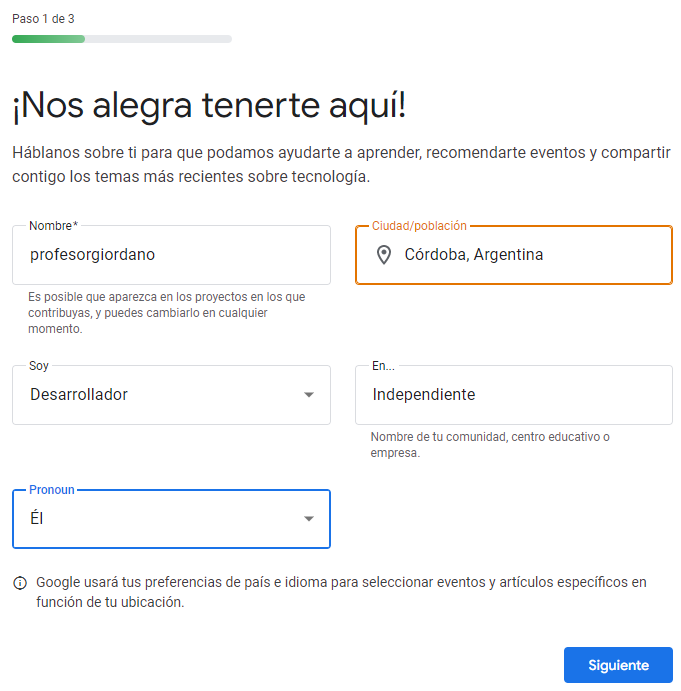


1. Copié la dirección de Heroku donde está el backend:

<https://ap-rag-backend.herokuapp.com/>

1. En Chrome me logueé como profesorgiordano@gmail.com
2. Entré al sitio de Firebase y en las opciones puse “Crear perfil de desarrollador”







1. Volví a Firebase y creé un nuevo proyecto, de nombre RAG-AP-FrontEnd
2. Instalé Firebase tools con el comando npm install -g firebase-tools en una nueva terminal de Visual Studio Code.
3. Aparecen algunos errores:

PS D:\PROYECTOS\ARGENTINA PROGRAMA\portfolio-Frontend> npm install -g firebase-tools

npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random

for details.

npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)

npm WARN deprecated har-validator@5.1.3: this library is no longer supported

npm ERR! code ERR\_SOCKET\_TIMEOUT

npm ERR! network Socket timeout

npm ERR! network This is a problem related to network connectivity.

npm ERR! network In most cases you are behind a proxy or have bad network settings.

npm ERR! network

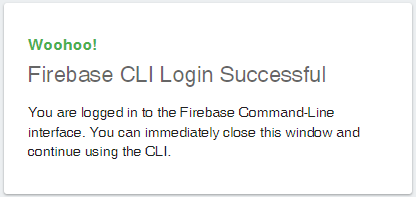
npm ERR! network If you are behind a proxy, please make sure that thenpm ERR! network 'proxy' config is set properly. See: 'npm help config'

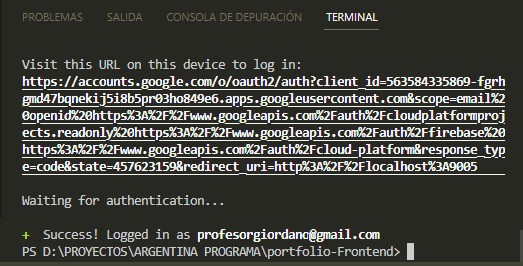
npm ERR! A complete log of this run can be found in:

npm ERR! C:\Users\Usuario\AppData\Local\npm-cache\\_logs\2022-10-31T23\_29\_50\_804Z-debug-0.log

PS D:\PROYECTOS\ARGENTINA PROGRAMA\portfolio-Frontend>

1. Lo volví a ejecutar y anduvo (con algunos warnings).
2. Me logueé con el comando firebase login (Abrió un navegador para loguearme con la cuenta de Google)





1. Ejecuté el comando ng build para compilar el Frontend
2. Inicié el proyecto con el comando firebase init
3. Ejecuté el comando firebase deploy. La primera vez me dio error de conexión. La segunda vez anduvo.
4. La dirección del frontend es:

<https://rag-ap-frontend.web.app>

1. Genero los usuarios con Wordbench
2. Completo las tablas de los usuarios con postman

{

    "nombre":"admin",

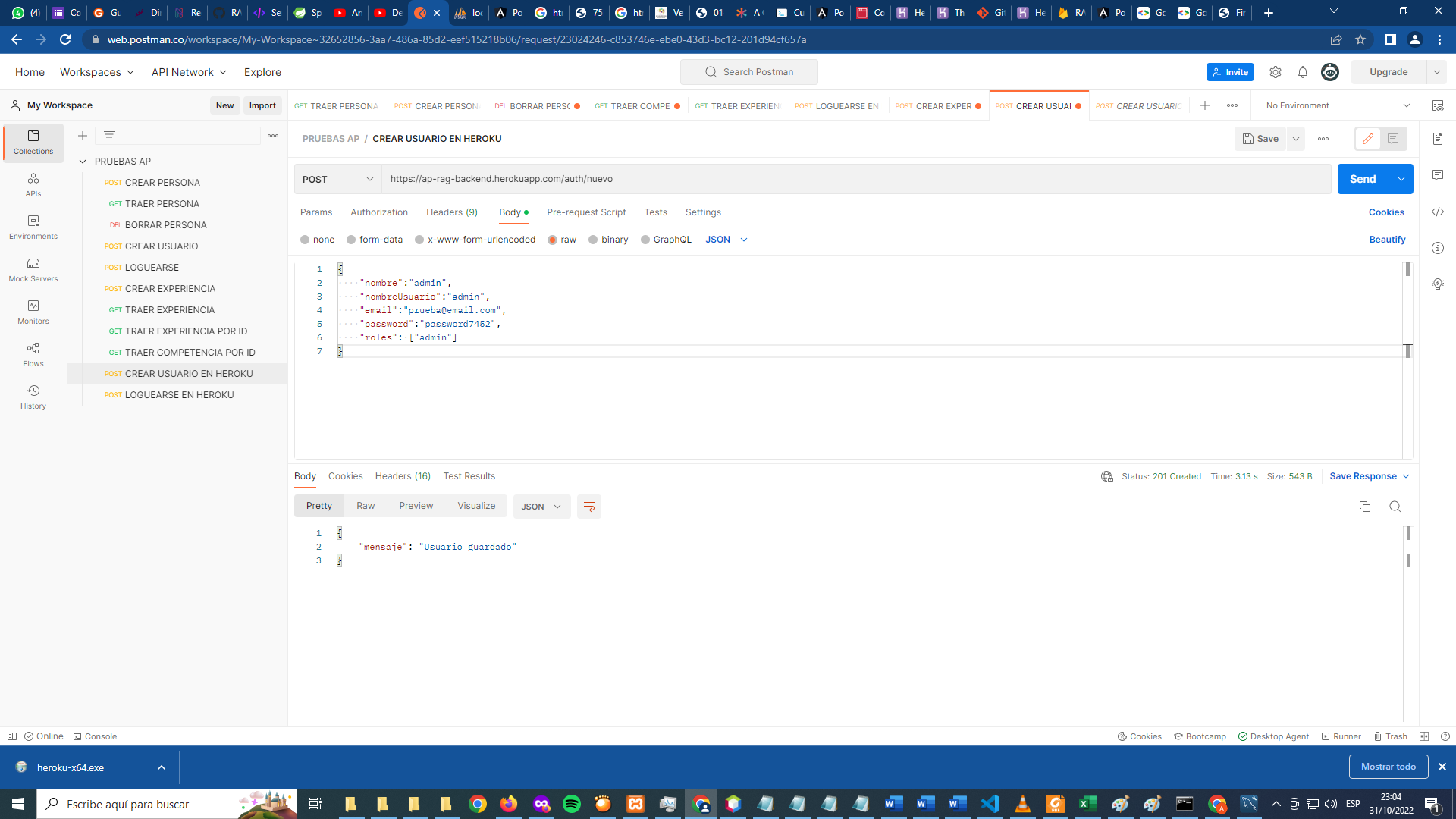
    "nombreUsuario":"admin",

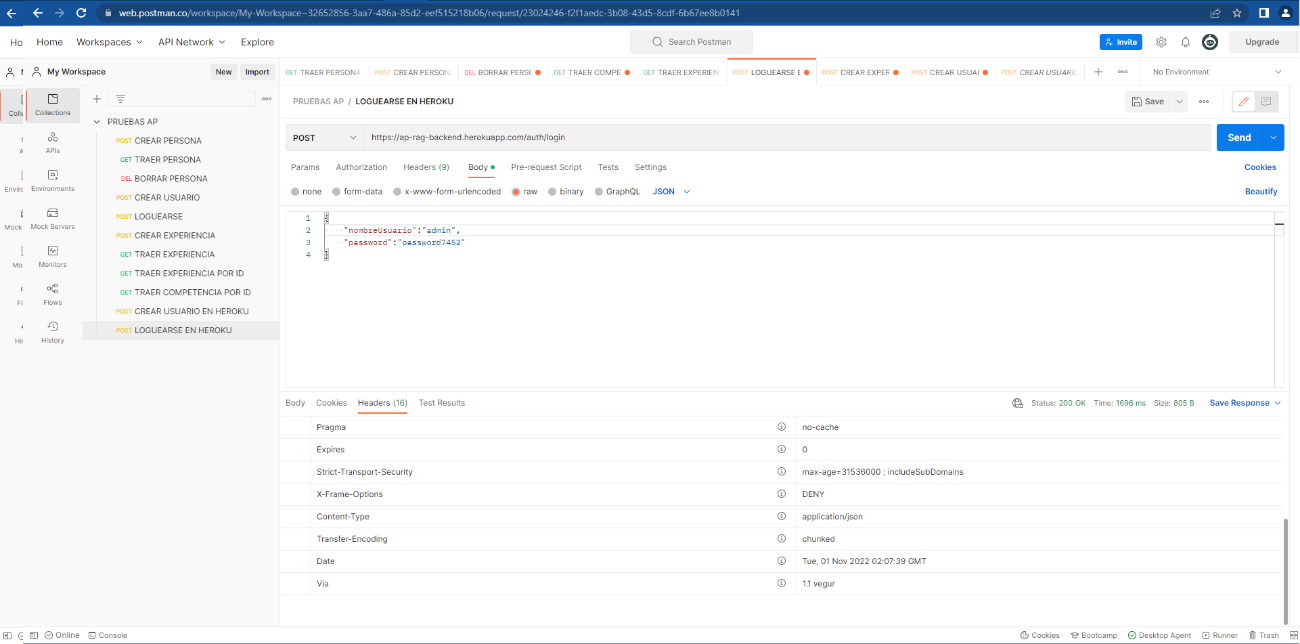
    "email":"prueba@email.com",

    "password":"password7452",

    "roles": ["admin"]

}





{

    "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsImlhdCI6MTY2NzI2ODQ1OSwiZXhwIjoxNjY3MjcyMDU5fQ.HqtY-wmSW9P96ME3grlzU4qVLD2o9S4G8NyFsgmBzDp6TcN\_PWwE1tyhDad1E7ZvS5ysFDK3m7RY4Sse66R0gA",

    "bearer": "Bearer",

    "nombreUsuario": "admin",

    "authorities": [

        {

            "authority": "ROLE\_USER"

        },

        {

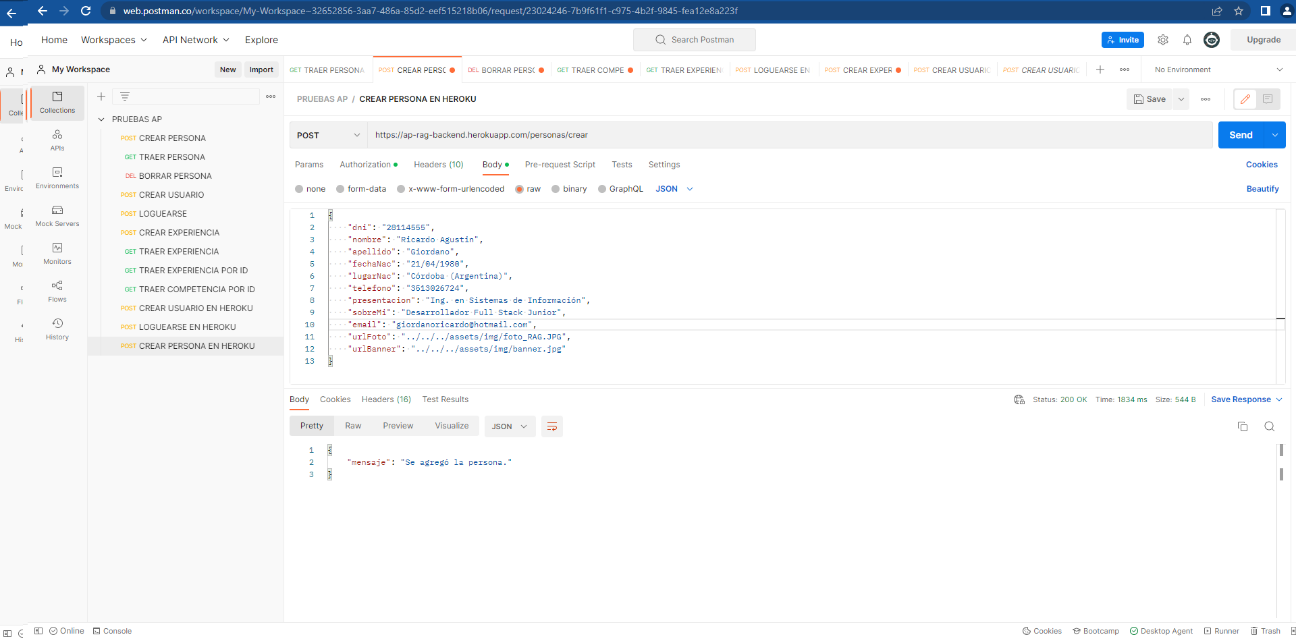
            "authority": "ROLE\_ADMIN"

        }

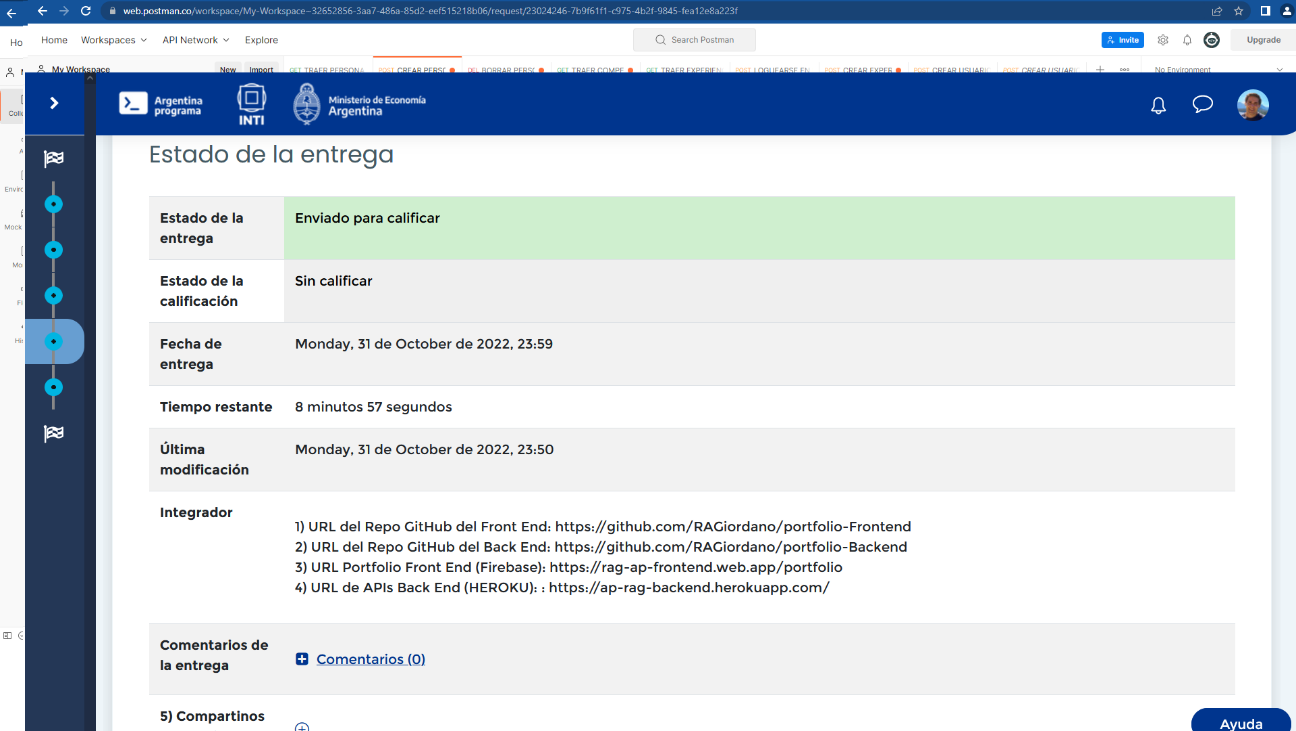
    ]

}

1. Copié el token y lo usé para crear una persona



1. Entregué el proyecto



**01/11/2022**

1. Moví el proyecto escrito y este documento de registro paso a paso a la carpeta ***portfolio-Backend\DOCUMENTOS***