

Sres. Argentina Programa

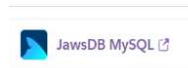
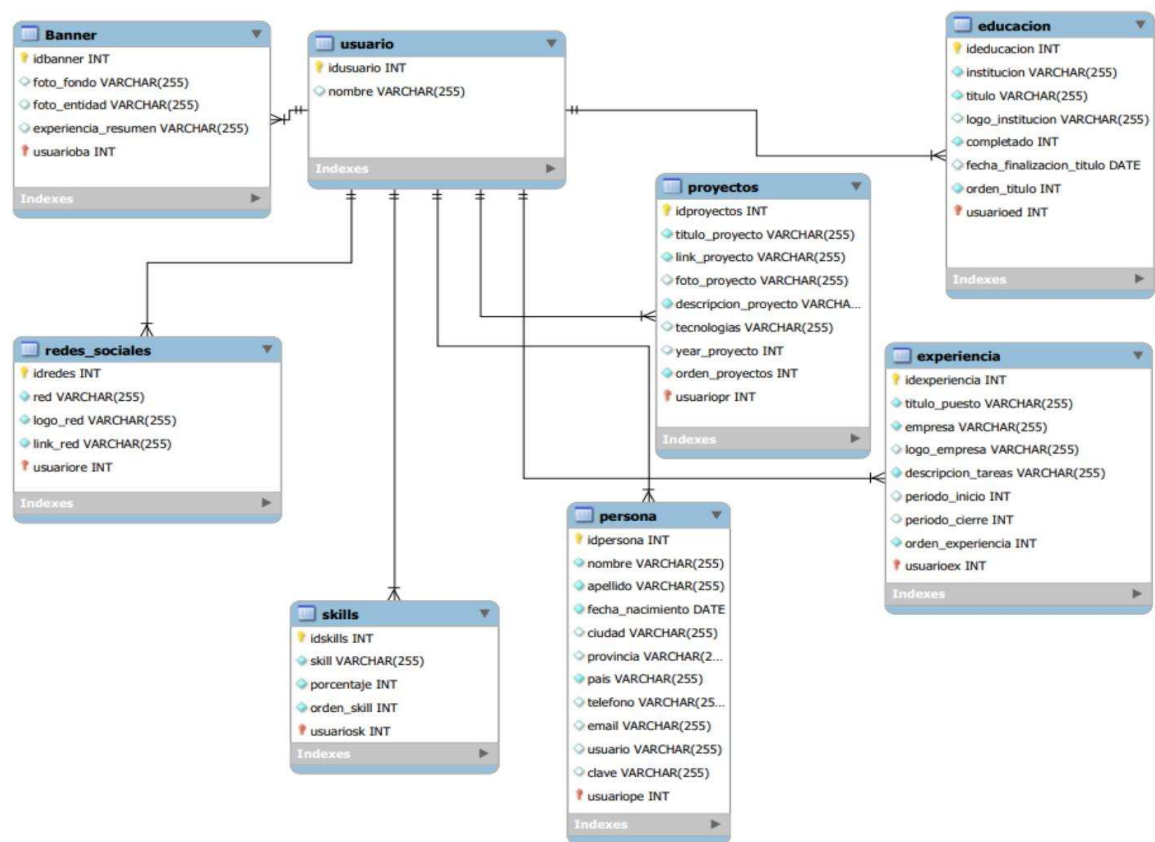
Ministerio de Producción de la Nación Argentina

S / D

Mi nombre es Pedro Damián Bellesi, DNI 24.392.039 y me dirijo a Uds. A efectos de presentar un resumen sobre el desarrollo de mi portfolio personal web para el curso YO PROGRAMA. Para ello divido la información de acuerdo los pasos que lleve para a su planificación y desarrollo, primero diseñando la base de datos en SQL, posteriormente desarrollando la API del Back End, y por último del Front End.

Base de Datos

Utilización de MySql, para la base de datos relacional. A continuación, el diagrama de relaciones de la base de datos:



Para la implementación Online se utilizó un Plugin de Heroku :

A continuación, se remiten modelo de los End Point de la Base de datos utilizados con Postman para cargar algunos datos de la base de datos.

Tabla usuarios:

POST: <https://pedrobelleportfolio.herokuapp.com/api/usuario/mas>

GET: <https://pedrobelleportfolio.herokuapp.com/api/usuario/ver>

```
{"idusuario": 1, "nombre": "Pedro"}
```

Tabla persona:

POST: <https://pedrobellesiportfolio.herokuapp.com/api/persona/mas>

GET: <https://pedrobellesiportfolio.herokuapp.com/api/persona/ver>

```
{  "idpersona": 0,  "nombre": "Pedro Damian",  "apellido": "Bellesi",  "fecha_nacimiento": "1975-06-03T00:00:00.000+00:00",  "ciudad": "General Roca",  "provincia": "Rio Negro",  "pais": "Argentina",  "telefono": "541124978888",  "email": "pedrobellesi@gmail.com",  "usuarioclave": null,  "clave": null,  "usuariope": {    "idusuario": 1,    "nombre": "Pedro"  } }
```

Tabla banner:

POST: <https://pedrobellesiportfolio.herokuapp.com/api/banner/mas>

GET: <https://pedrobellesiportfolio.herokuapp.com/api/banner/ver>

```
{  "idbanner": 0,  "foto_fondo": "./assets/prog3b.jpg",  "foto_entidad": "./assets/pedro_foto.jpg",  "experiencia_resumen": "Full Stack Developer Jr., Contador Público Nacional, Martillero, Corredor Inmobiliario. Entusiasta, colaborativo, emprendedor y con gran interés por la mejora y la resolución de problemas.",  "usuarioba": {    "idusuario": 1,    "nombre": "Pedro"  } }
```

Tabla redes_sociales:

POST: <https://pedrobellesiportfolio.herokuapp.com/api/redsocial/mas>

GET: <https://pedrobellesiportfolio.herokuapp.com/api/redsocial/ver>

```
{  "idredes": 0,  "red": "facebook",  "logo_red": "1",  "link_red": "#",  "usuariore": {    "idusuario": 1,    "nombre": "Pedro"  } }
```

Tabla educacion:

POST: <https://pedrobellesiportfolio.herokuapp.com/api/educacion/mas>

GET: <https://pedrobellesiportfolio.herokuapp.com/api/educacion/ver>

DELETE: <https://pedrobellesiportfolio.herokuapp.com/api/educacion/borra>

```
{  "ideducacion": 0,  "institucion": "Universidad Nacional del Comahue",  "titulo": "Contador Publico Nacional",  "logo_institucion": "./assets/ucomahue.png",  "completado": 1,  "fecha_finalizacion_titulo": "1998-05-05T00:00:00.000+00:00",  "orden_titulo": 1,  "usuarioed": {    "idusuario": 1,    "nombre": "Pedro"  } }
```

Tabla experiencia:

POST: <https://pedrobellesiportfolio.herokuapp.com/api/experiencia/mas>

GET: <https://pedrobellesiportfolio.herokuapp.com/api/experiencia/ver>

DELETE: <https://pedrobellesiportfolio.herokuapp.com/api/experiencia/borra>

```
{  "idexperiencia": 0,  "titulo_puesto": "Auditor Interno",  "empresa": "Intercargo SAC (Aeropuerto internacional de Ezeiza y escalas)",  "logo_empresa": "",  "descripcion_tareas": "Auditoria Integral en UAI empresa de Rampa. Presentación de Informes. Control Presupuestario. Redacción de Normas. Elaboración de índices \n y ratios. Auditoria de Sistemas Informáticos.",  "year_inicio": 2000,  "year_cierre": 2005,  "orden_experiencia": 0,  "usuarioex": {    "idusuario": 1,    "nombre": "Pedro"  } }
```

Tabla skills:

POST: <https://pedrobellesiportfolio.herokuapp.com/api/skill/mas>

GET: <https://pedrobellesiportfolio.herokuapp.com/api/skill/ver>

DELETE: <https://pedrobellesiportfolio.herokuapp.com/api/skill/borra>

```
{  "idskills": 0,  "skill": "SQL",  "porcentaje": 85,  "orden_skill": 0,  "usuarios": {  "idusuario": 1,  "nombre": "Pedro"  } }
```

Tabla proyectos:

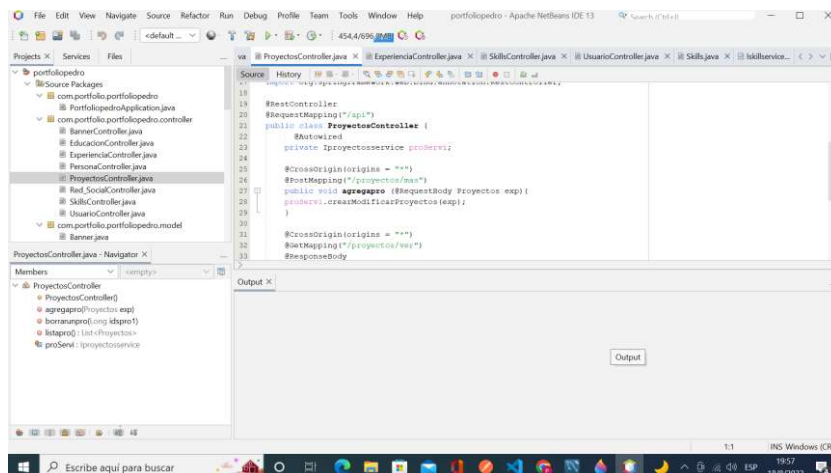
POST: <https://pedrobellesiportfolio.herokuapp.com/api/proyectos/mas>

GET: <https://pedrobellesiportfolio.herokuapp.com/api/proyectos/ver>

DELETE: <https://pedrobellesiportfolio.herokuapp.com/api/proyectos/borra>

BACK END

EL Back End se instaló sobre Heroku, siendo su dirección de implementación: <https://pedrobellesiportfolio.herokuapp.com>. El proyecto fue realizado inicialmente en Apache Net Beans 13, de la cual se presenta copia de pantalla de su realización:



Se utilizó lenguaje Java y Spring Boot para realizar el proyecto, de acuerdo a lo enseñado en el material del curso. Cabe señalar que como dificultades que se me presentaron fueron:

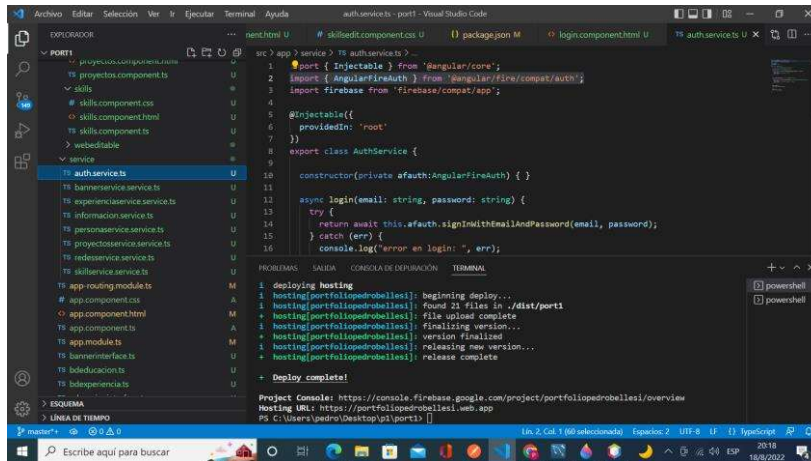
- Los puertos que utilizaban las bases de datos de mi PC se superponían, así que debí originalmente asignar otros puertos a los que vienen por Default.
- Bloqueo de seguridad del servidor apache, por lo que debí buscar en internet forma de que la API permita solicitar información de otras IP. Para ello se utilizó la anotación: `@CrossOrigin(origins = "*")`
- Implementación de "Many to one" en la persistencia de Spring Boot y así seguir con lo proyectado en el diseño de las bases de datos, que no se explicó en el material del curso. Por lo que debí investigar bastante y solicitar información a mi profesor del curso (que me lo facilitó). Con ello pude resolverlo satisfactoriamente.

Una copia del back end del porfolio se encuentra en GITHUB:

<https://github.com/PedroBellesiPortfolio/Backend>

FRONT END

El Front End se instaló sobre Firebase, siendo su dirección de implementación: <http://portfoliopedrobellesi.firebaseio.com>. El proyecto fue realizado utilizando Visual Studio Code, de la cual se presenta copia de pantalla de su realización:



Se utilizó lenguaje JavaScript, TypeScript con Angular, HTML, CSS y Bootstrap para realizar el proyecto, de acuerdo a lo enseñado en el material del curso. Se trató de ser fiel inicialmente al modelo ejemplo del proyecto, a los efectos de implementar un DOM que cumpla con los requisitos propuestos. Cabe señalar que, como característica particular del diseño e implementación, afectos de comprender el funcionamiento completo del portofio y comprender las tecnologías utilizadas se efectúa una breve reseña de los mismos:

- La web como se informó fue desarrollada utilizando Bootstrap 5 para el diseño. Por lo que conviné el CSS tradicional, con el sistema de grillado y definición de clases de Bootstrap.
- La web fue efectuada en Angular, utilizando un sistema de SPA, sistema de routing y sistema de acceso para componentes de edición del contenido del portofio.
- En cuanto a la definición de los componentes en Angular, intente tratarlos en lo posible como objetos independientes, desde el la SPA a un simple botón, con el objetivo de tratar de reutilizar código. Si veía que la dificultad de su programación para discriminarlo como un componente requería excesiva y compleja implementación de código (por ejemplo, la comunicación entre componentes, padre, hijo, nieto, etc, debido a que con las herramientas que recién estoy implementando y comprendiendo, se me podía dificultar), lo veía como un límite en la discriminación y determinación de objetos. Por lo cual trate de ser eficaz y no tan eficiente, pero cumplir con el objeto del trabajo, que es entregar un portofio de acuerdo a lo solicitado.
- Para la comunicación con la API REST, se utilizaron servicios independientes para cada tabla, cumpliendo con la comunicación y la discriminación de los controller de la API. Cabe señalar con respecto a la repetición de código, en un futuro se podría mejorar con la implementación de patrones de diseño, pero en esta ocasión lo hice en forma discriminada para comprender mejor el código y ser mucho más simple la programación.

- Respecto al sistema de seguridad de la web, se implementó el sistema de autenticación de firebase y su sistema de guards. Habría sido mucho más segura una web con un sistema de seguridad JWT, pero como mi profesor del curso me informó no era condicionante del portfolio, sino que era sugerido. Y como inicialmente no se incluyó en el material del curso (nuestro profesor nos facilitó material ante la solicitud de los compañeros del curso) sobre como implementar la programación del JWT del lado de la API con Spring Boot, me decidí por el uso de guards y autenticación con firebase.
- Respecto a la implementación de formulario se utilizó el sistema de formularios reactivos tal cual es solicitado en el curso. Se incluyo algunos validadores, los meramente necesarios a los efectos de evitar enviar a la API registros con error (Por ejemplo, un string en el lugar donde la base de datos solicita un numero).
- La edición del orden de los registros ingresados es mediante el sistema Drag and Drop de Angular: **@angular/cdk/drag-drop** module. Cada vez que se cambia el orden de un registro se actualiza la base de datos.
- Cabe señalar el las bases de datos gratuitas sugeridas si bien son rápidas, la primera vez que se llama a la API REST es probable que no cargue los datos y se deba actualizar la web para que termine de cargarlos. Por ello, se intentó mejorar el sistema de espera con el uso de observables, promesas y hasta se implementó un ngfor (`<div *ngFor="let item of [].constructor(3600); let x = index">`) con conteo para retardar la solicitud de cada componente por el DOM y así darle tiempo al sistema para la carga de los datos (completos y con el orden que se le asigno en la edición de los registros de cada componente).
- Se implementó un sistema de seguridad en el eliminado de registros, sobre el front end, para evitar que mientras se espera que se refresque en el DOM un registro recientemente eliminado, no pueda ser nuevamente clickeado, ocasionado una solicitud que pueda causar problemas a la persistencia de la API. Para ello agrego una confirmación y que no puedan eliminarse los registros recientemente eliminados.
- Se implementó un sistema de visualización de links de redes sociales en los que, si no se agrega ningún link a alguna de las mismas, desaparece la red del DOM, debiendo ingresarse a la edición de redes sociales y agregar un link a la misma para que aparezca nuevamente sobre el DOM.
- Para algunas fotos se incluyó la foto por defecto, en caso de que en la edición se hubiese indicado un link a una foto inexistente, a fin de evitar la desprolijidad en el diseño que se visualiza.

Una copia del front end del portfolio se encuentra en GITHUB:

<https://github.com/PedroBellesiPortfolio/frontend>

CONCLUSION

EL desarrollo del presente portfolio me permitió profundizar y aprender sobre el desarrollo de un Full Stack. Durante el proceso que me llevo el desarrollo, no solo pude introducirme y comprender las tecnologías utilizadas, sino que me permitió crecer en lo profesional y en lo personal. Por lo que me encuentro plenamente agradecido a todos aquellos que integran Argentina Programa, por permitirme tener acceso en forma totalmente gratuita a este aprendizaje.

Desde ya, quedo a su disposición para consultas sobre bugs en la web o la API, o formalismos de su desarrollo, sino también de surgir el interés para charlas o presentaciones como estudiante del programa.

También pongo a disposición el presente portfolio para que sea ingresado en las bases de datos de empresas a efectos de que sea consultado por aquellas que demanden programadores.

Atte.

Pedro Damián Bellesi

