



# TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES WEB

Departamento de Informática

## PROYECTO

QRiculum

**Manual Técnico**

Autor/es: Ana Larios Vettoretti

# Índice

- **Introducción**
- **Arquitectura de la aplicación**
  - **Frontend**
    - **Tecnologías usadas**
    - **Entorno de desarrollo**
  - **Backend**
    - **Tecnologías usadas**
    - **Entorno de desarrollo**
- **Documentación técnica**
  - **Análisis**
  - **Desarrollo**
  - **Pruebas realizadas**
- **Proceso de despliegue**
- **Propuesta de mejoras**

- **Bibliografía**

- **Introducción**

QRiculum es un proyecto web destinado a personas demandantes de empleo que facilita la presentación de currículums a empresas, dotando a cada usuario de un QR único cuyo escaneo muestra la página personalizada con el currículum de dicho usuario.

- **Arquitectura de la aplicación**

La arquitectura de mi aplicación (como se detalla en el diagrama de la documentación técnica) tiene como objetivo separar por completo la parte cliente de la parte servidor, pudiendo utilizar la API-REST desde cualquier tipo de dispositivo, desde una aplicación web a una en Android o cualquier tipo de cliente.

- **Frontend**

- **Tecnologías usadas**

- Angular 13: Lo he escogido por la novedad y la posibilidad de trabajar con una SPA. Además me parece una buena elección al ser una de las tecnologías más utilizadas actualmente en el desarrollo de aplicaciones web

- Angular Material

- Bootstrap 5

- **Entorno de desarrollo**

- Visual Studio Code 1.68: He escogido este entorno de desarrollo porque es con el que estoy familiarizada y me facilita el trabajo con las extensiones

- **Librerías**

- Ngx-Toast: Notificaciones

- Ngx-Avatars: Generar avatares con las iniciales

- Angular2-qrcode: Generar códigos QR

- SweetAlerts2: Modales de notificaciones

- **Backend**

- **Tecnologías usadas**

- Java 18: He escogido Java debido a que es un lenguaje con el que estoy familiarizada y a que es uno de los más utilizados actualmente.

- Spring Boot: He escogido Spring Boot debido a la rapidez y la ayuda que proporciona con las anotaciones, la posibilidad de desarrollar con facilidad microservicios, además de facilitar la configuración de dependencias y el despliegue del servicio.

- MongoDB: He escogido MongoDB debido a la gran flexibilidad que permite, debido a que en mi caso, cada usuario puede tener un número variable de

- **Entorno de desarrollo**

- Eclipse 2022/06: He escogido este entorno de desarrollo porque lo había utilizado anteriormente con Java y tiene un buen rendimiento.

- MongoDBCompass: Este programa no es un entorno de desarrollo como tal pero sirve para administrar la base de datos .

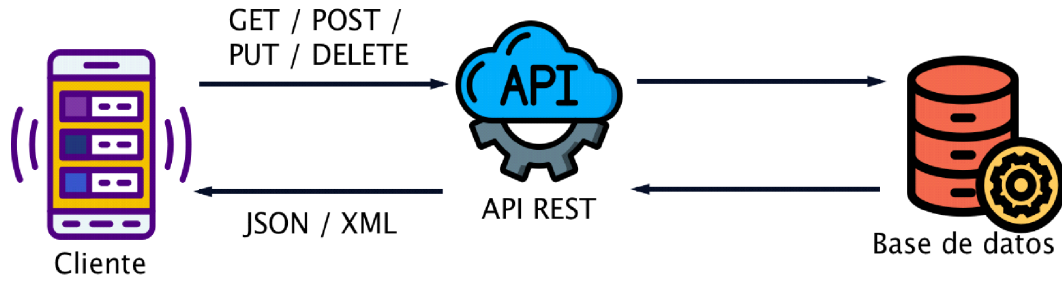
- **Pruebas**

- Postman: Es un programa con el que se pueden hacer pruebas de peticiones a la API y comprueba las respuestas y la conexión con el servidor

- **Documentación técnica**

- **Análisis**

Arquitectura de la aplicación web



### Diagrama de clases

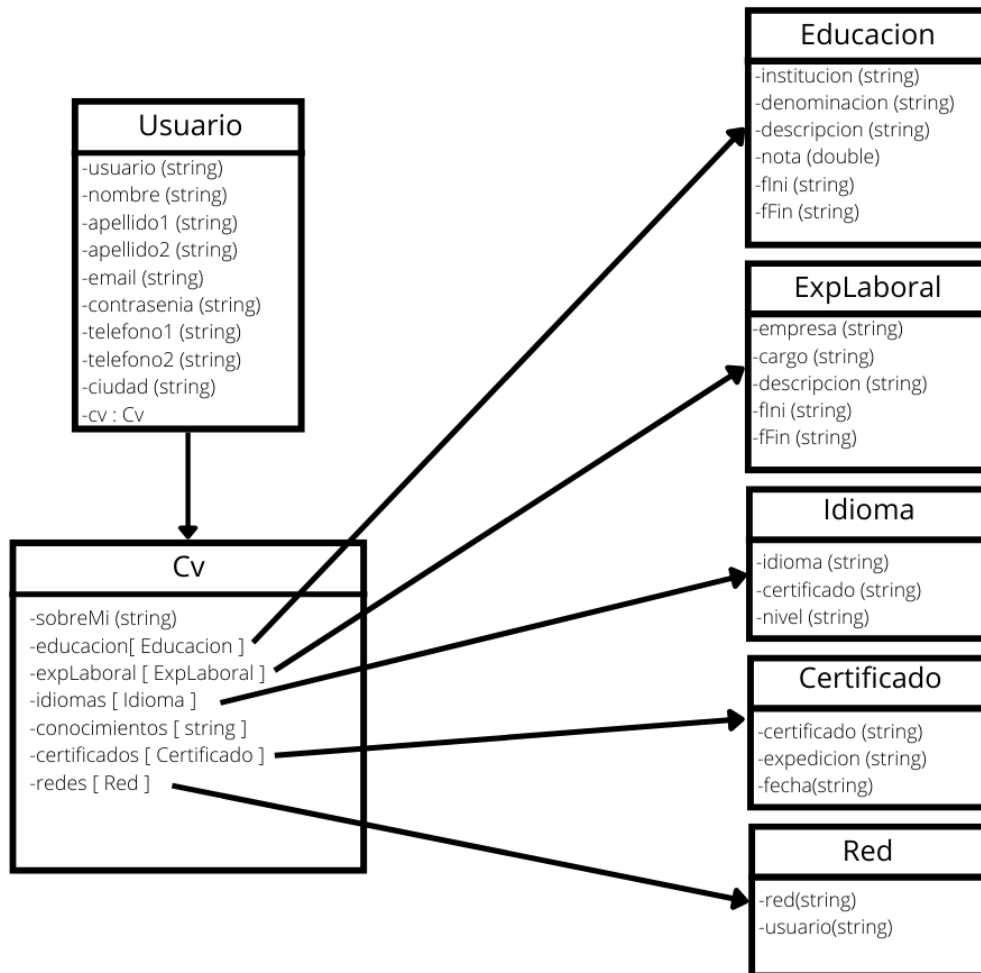
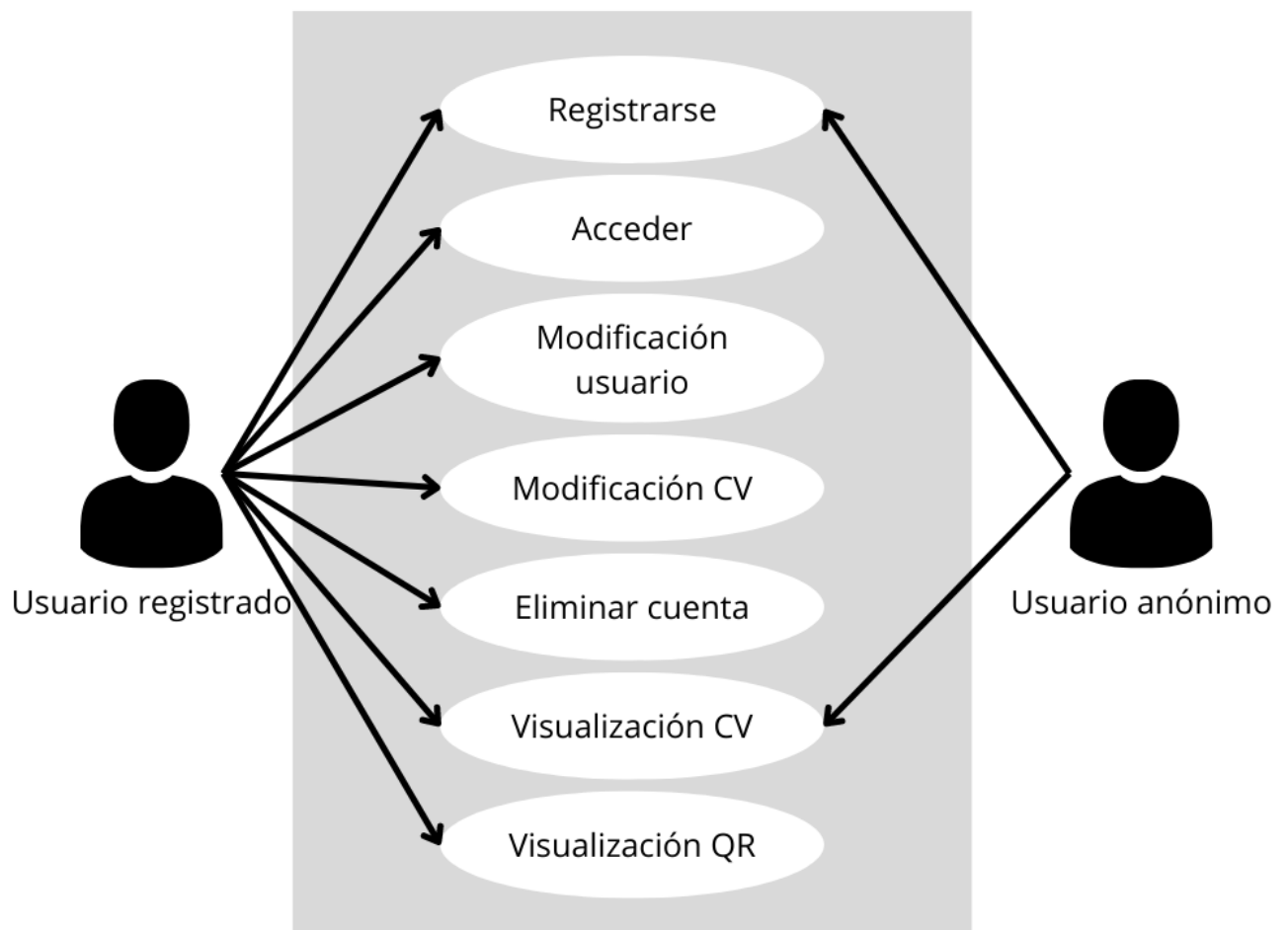


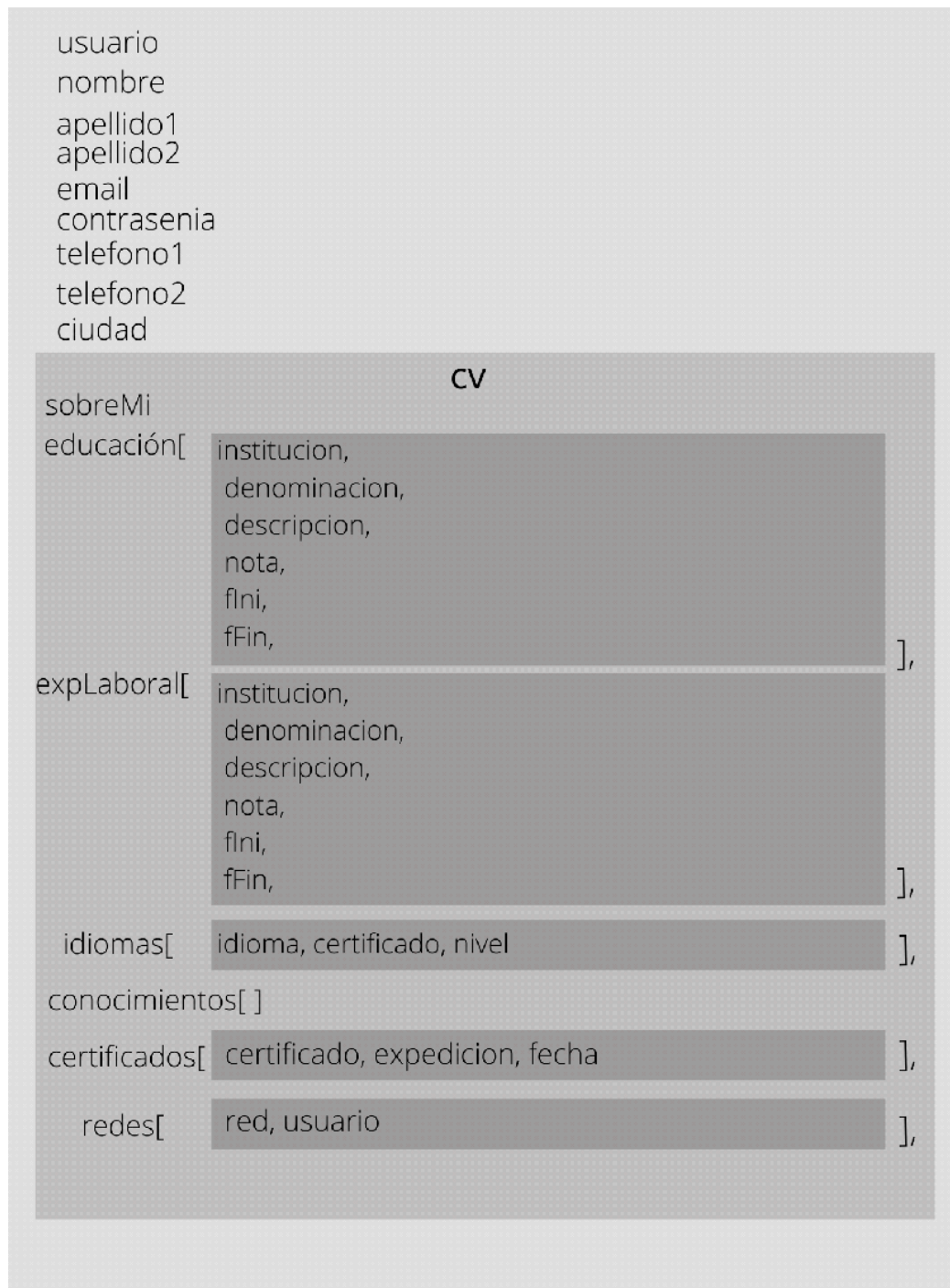
Diagrama de casos de uso



- **Desarrollo (diagrama de secuencias,...)**

## Diagrama de la base de datos MongoDB

### usuario



- **Pruebas realizadas**



Las pruebas realizadas han sido de dos tipos:

- Pruebas manuales:

Testeo manual del correcto funcionamiento, introducción de datos, pruebas de diferentes supuestos, etc.

- Pruebas mediante programa:

Pruebas de comunicación entre base de datos y la API REST realizadas con el programa Postman, que permite hacer consultas mediante URL y analizar la respuesta del servidor y su conexión con la base de datos.

## • Proceso de despliegue

El proceso de despliegue se ha llevado a cabo utilizando diferentes hostings:

- **Base de datos** - MongoDB Atlas Database

Se puede crear un usuario y crear bases de datos de forma gratuita. Permite el acceso sólo a las IP seleccionadas en la versión gratuita y permite acceder mediante un enlace. En el caso de la conexión con Spring Boot en mi caso sería agregando esta línea al archivo application.properties:

```
spring.data.mongodb.uri=mongodb+srv://iesalbarregas:iesalbarregas@qriculum.ekllq.mongodb.net/
```

- **API-REST** - Heroku

Se crea un usuario gratis y permite desplegar la aplicación vinculando la cuenta de GitHub, existe la opción de hacer despliegues automáticos con cada push al repositorio. El enlace para acceder a la aplicación desplegada es:

<https://qriculum-back.herokuapp.com>

Un ejemplo de uso en el que se obtienen todos los usuarios de la BBDD sería:

<https://qriculum-back.herokuapp.com/usuarios>

- **Front-end** - Vercel

Se crea un usuario gratuito y también permite conectar la cuenta de GitHub y realizar despliegues automáticos con cada push del repositorio. La conexión con la API-REST se realiza mediante las peticiones HTTP, así que bastaría con indicar la url donde se encuentra el servidor con su correspondiente endpoint.

<https://qriculum-gquidgoja-analariosvettoretti.vercel.app>

## • **Propuestas de mejoras**

Las mejoras propuestas son las siguientes:

- Implementación del buscador para poder buscar usuarios por conocimientos (Angular, MongoDB, etc). Daría la oportunidad de conectar con personas que posean habilidades concretas para un proyecto o puesto de trabajo.
- Posibilidad de elegir entre varios diseños para mostrar el CV: permitiría tener más opciones de personalización en cuanto a colores, diseños, etc
- Posibilidad de ver el número de visitas al CV: para saber la cantidad de personas que ha visualizado el currículum.

## • **Bibliografía**

- <https://angular.io/docs>
- <https://material.angular.io/guides>
- <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
- <https://devcenter.heroku.com/articles/deploying-spring-boot-apps-to-heroku>