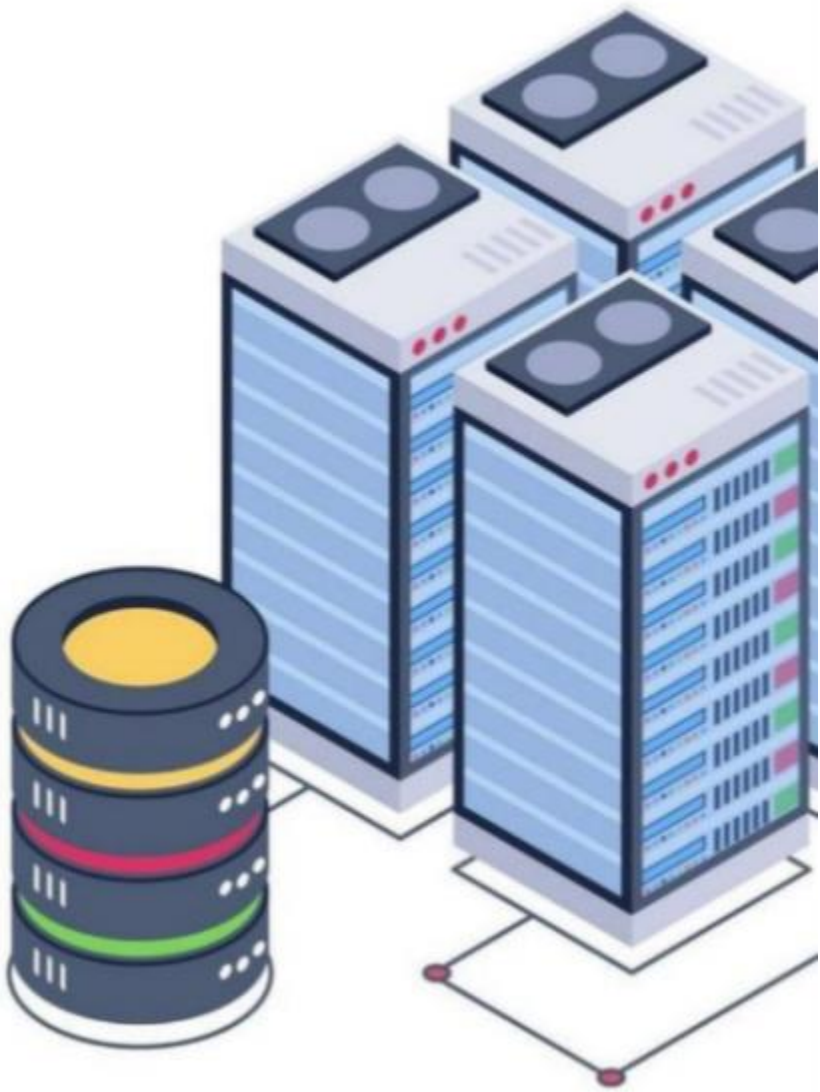
PROYECTO FINAL
SEGUNDA ENTREGA



Programación de servicios web.

Realizado por:

Juan Ramón Rodríguez Armas.

Fernando Corona Preciado.

Jesús Axel García De la cruz.

Carlos Samuel Fierros García.

Olaf Alonso Martínez

Breve descripción.

Este proyecto nace de la necesidad de mejorar nuestros entornos urbanos, con una aplicación web que permita a las personas levantar reportes a las autoridades correspondiente sobre distintas problemáticas en nuestro entorno, como levantar un reporte de basura acumulada a nuestro ayuntamiento, o fugas de agua al SIAPA o al CONAGUA, siendo principalmente problemas NO CRITICOS, ya que reportar un incendio por medio de una aplicación no sería tan eficiente como levantarlo por medio del 911. Creemos que permitir levantar reportes de forma rápida y simple a los usuarios, y que estas autoridades puedan responder a través de esta aplicación, se mejorara la estética y los entornos en los que vivimos.

Materias involucradas.

Las materias involucradas en este semestre serán las de bases de datos 2, ingeniería de software y programación web. La selección de estas materias es debido a todas estas aportan algo a nuestra formación, y desarrollar un proyecto que involucre estas materias, buscamos reforzar conocimientos que actualmente estamos aprendiendo y de lo aprendido anteriormente. La razon de escoger base de datos 2, es que es necesario guardar los reportes que hagan las personas en algún lugar y pasar la información a las autoridades correspondientes, guardar el contacto de las autoridades, ubicaciones, etc. Por parte de la parte de ingeniería de software, pues básicamente dar orden al proyecto y por programación web, queremos que esta app pueda ser accedida por casi cualquier dispositivo gracias al hacer la aplicación de manera web.

Objetivo ODS seleccionado.

Principalmente abarca la ODS 13 y 11, contribuir a la reducción de contaminación, o daños a cierto tipo de comunidades o en general ciudades, haciendo una identificación rápida y gestionando eficiente de problemas reportados por los mismos cuidados.

Objetivo y justificación del proyecto.

Objetivo: El uso facil de una aplicación sencilla para la comunidad involucrada para identificar problemas dentro de esa comunidad, y mejorar la respuesta de las autoridades correspondientes, y formar una organización responsable.

Justificación: La falta de una herramienta digital, ocasiona que muchos casos pasen mucho tiempo en el olvido, provocando una incomodidad hacia la comunidad. Un sistema como el que planeamos presentar permitirá mejorar esta gestión y dar una respuesta a dichos problemas, contribuyendo así a la sostenibilidad ambiental.

Tecnologías Por Utilizar

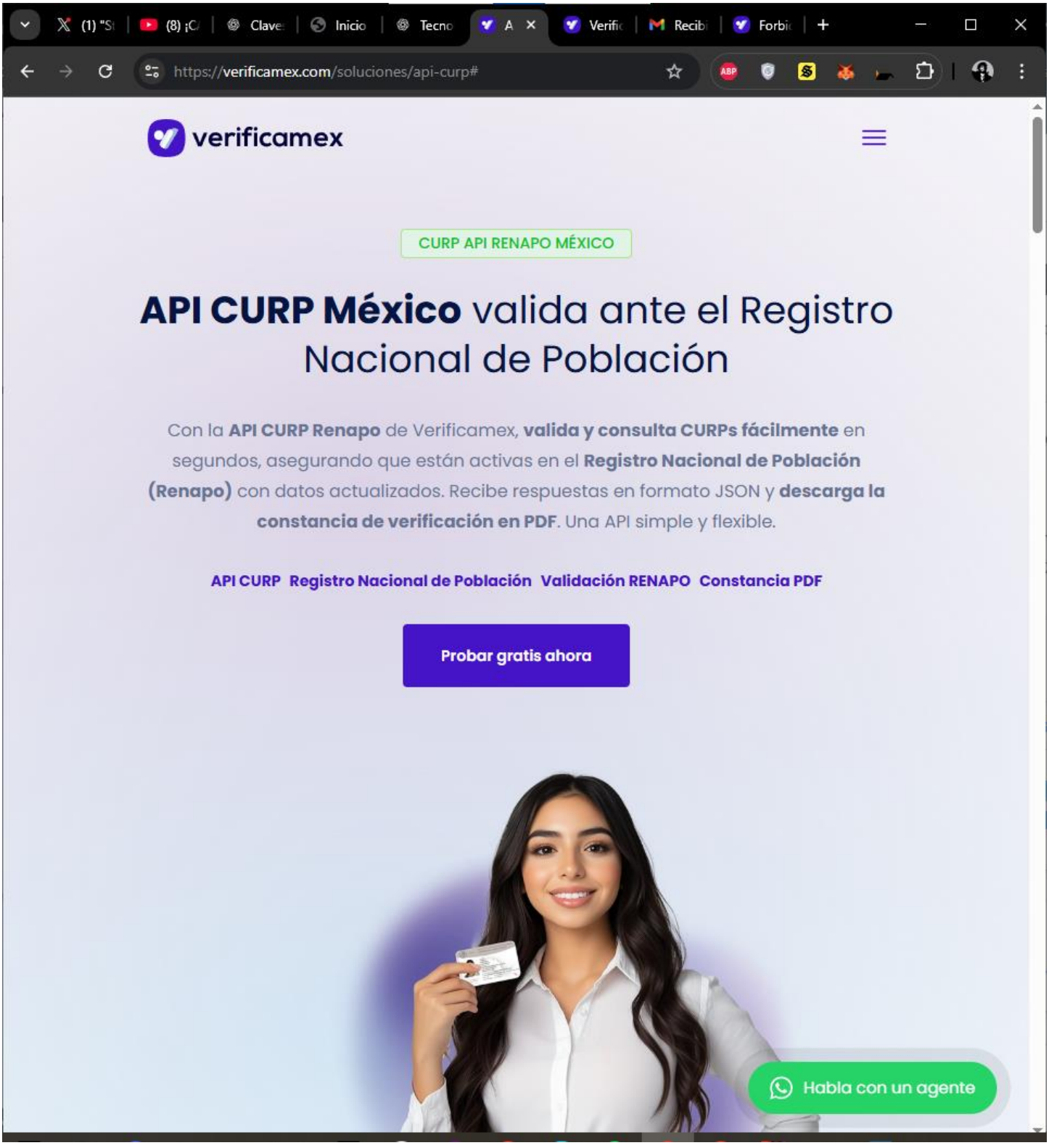
Las tecnologías que utilizaremos son diversas, ya que al tratarse de una aplicación web, es prácticamente obligatorio usar HTML para la estructura de la página, CSS para el diseño y JavaScript si queremos un sitio web dinámico. Estas serán, por tanto, las tecnologías empleadas en el desarrollo del Frontend.

En cuanto a la lógica de la aplicación, utilizaremos Java junto con el framework Spring Boot. Elegimos Java porque es un lenguaje que conocemos bien, además de ser versátil, seguro y ampliamente utilizado en el desarrollo del Backend de aplicaciones web.

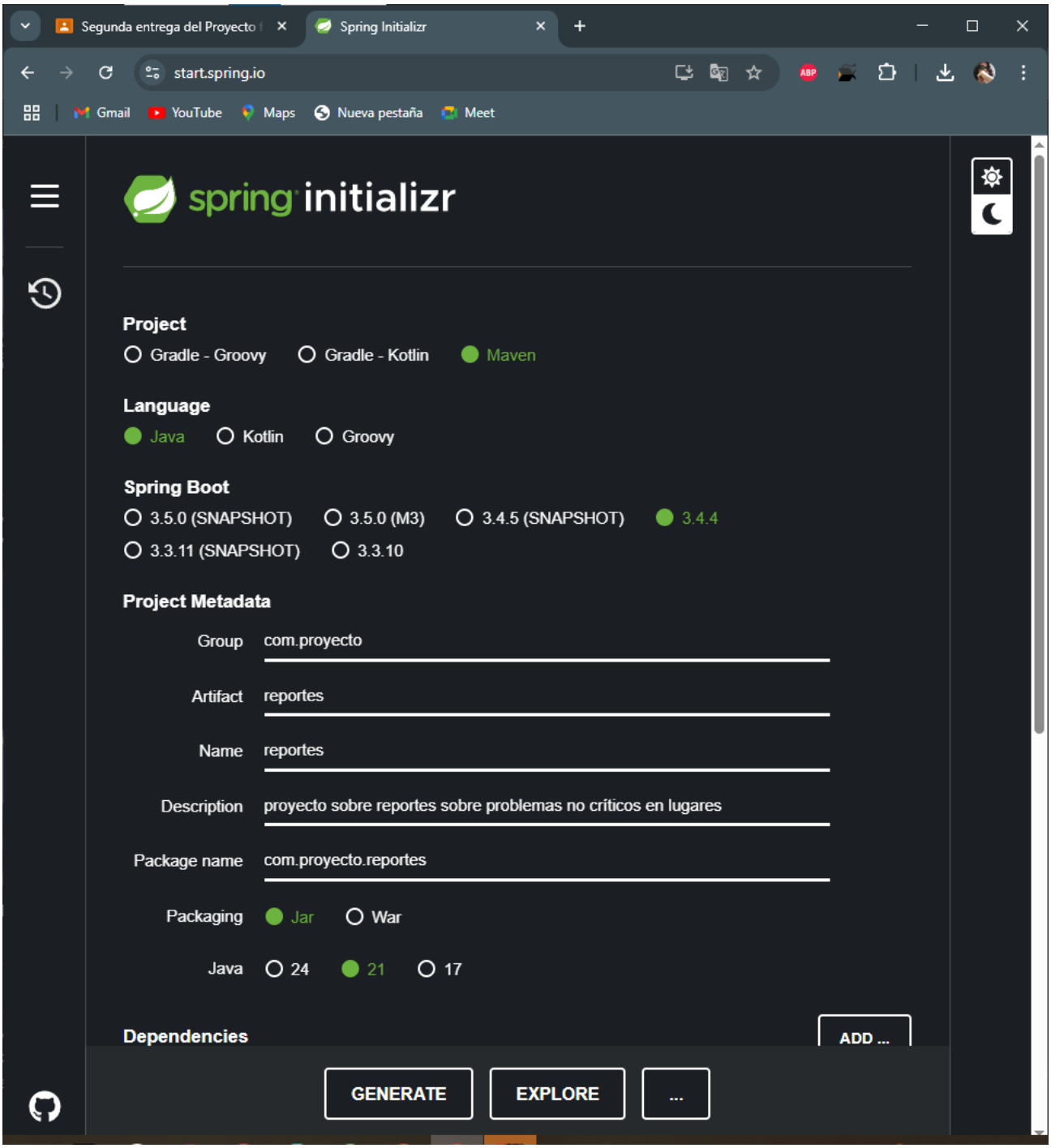
Optamos por Spring Boot porque nos permite crear aplicaciones web de forma mucho más rápida y eficiente que hacerlo desde cero. Este framework nos ofrece herramientas útiles como la inyección de dependencias mediante anotaciones (@), y, si se seleccionan las librerías adecuadas, también proporciona un servidor web embebido, como apache tomcat, y soporte para la creación de APIs RESTful. Estas APIs serán las encargadas de suministrar información al Frontend. Practicamente, Spring Boot nos brinda versatilidad y herramientas que nos permiten enfocarnos en la lógica de la aplicación, sin tener que preocuparnos por implementar todo desde cero.

En la parte de persistencia de datos, utilizamos MySQL, esto debido a que es sencillo de usar y muy eficiente, y debido a todo lo aprendido en el semestre, podríamos implementar triggers o procedimientos almacenados, para encapsular aun mas partes de la aplicación.

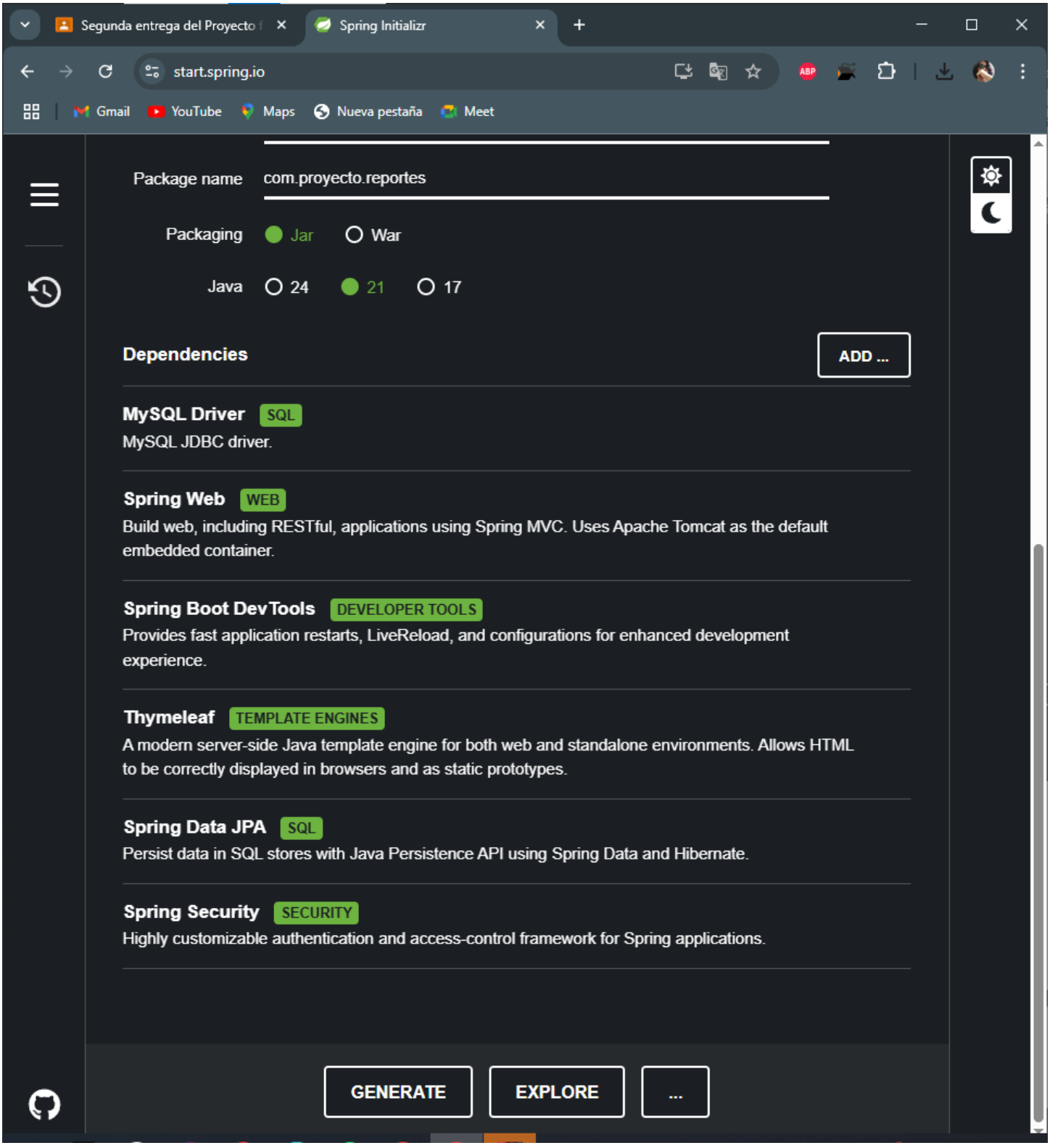
Para poder autenticar a los usuarios, y saber que son reales, se esta usando una autenticación por medio de la curp, donde utilizamos una API que nos puede confirmar si esta curp es valida o no, ya que consulta medios oficiales, y no solo es un algoritmo que calcularía la CURP.



Esta es el api que nos permite verificar si una CURP es real y que el usuario es real.



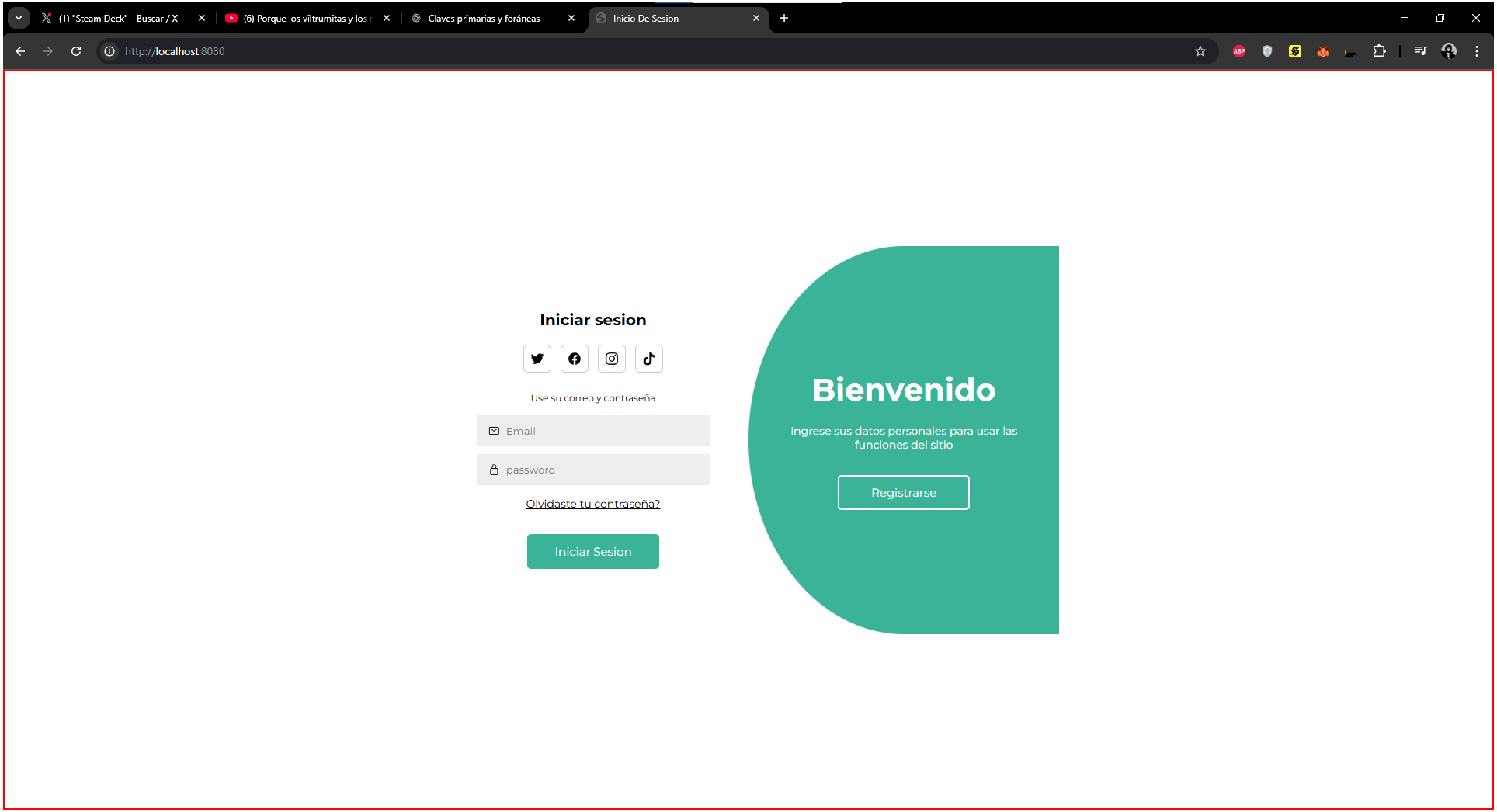
Aquí podemos ver una herramienta que nos proporciona la pagina web de Spring, que es un inizializador, donde elegimos el gestor de dependencias, que en nuestro caso fue Maven; el lenguaje a utilizar, que en este caso fue java; la versión de spring boot; la metadata del proyecto; como se guardara el proyecto, ya sea tipo Jar o War, y la versión de java a utilizar


Segunda entrega del Proyecto
Spring Initializr
start.spring.io
Gmail
YouTube
Maps
Nueva pestaña
Meet
Package name
com.proyecto.reportes
Packaging
Jar
War
Java
24
21
17
Dependencies
ADD ...
MySQL Driver
SQL
MySQL JDBC driver.
Spring Web
WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
Spring Boot DevTools
DEVELOPER TOOLS
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.
Thymeleaf
TEMPLATE ENGINES
A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.
Spring Data JPA
SQL
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
Spring Security
SECURITY
Highly customizable authentication and access-control framework for Spring applications.
GENERATE
EXPLORE
...

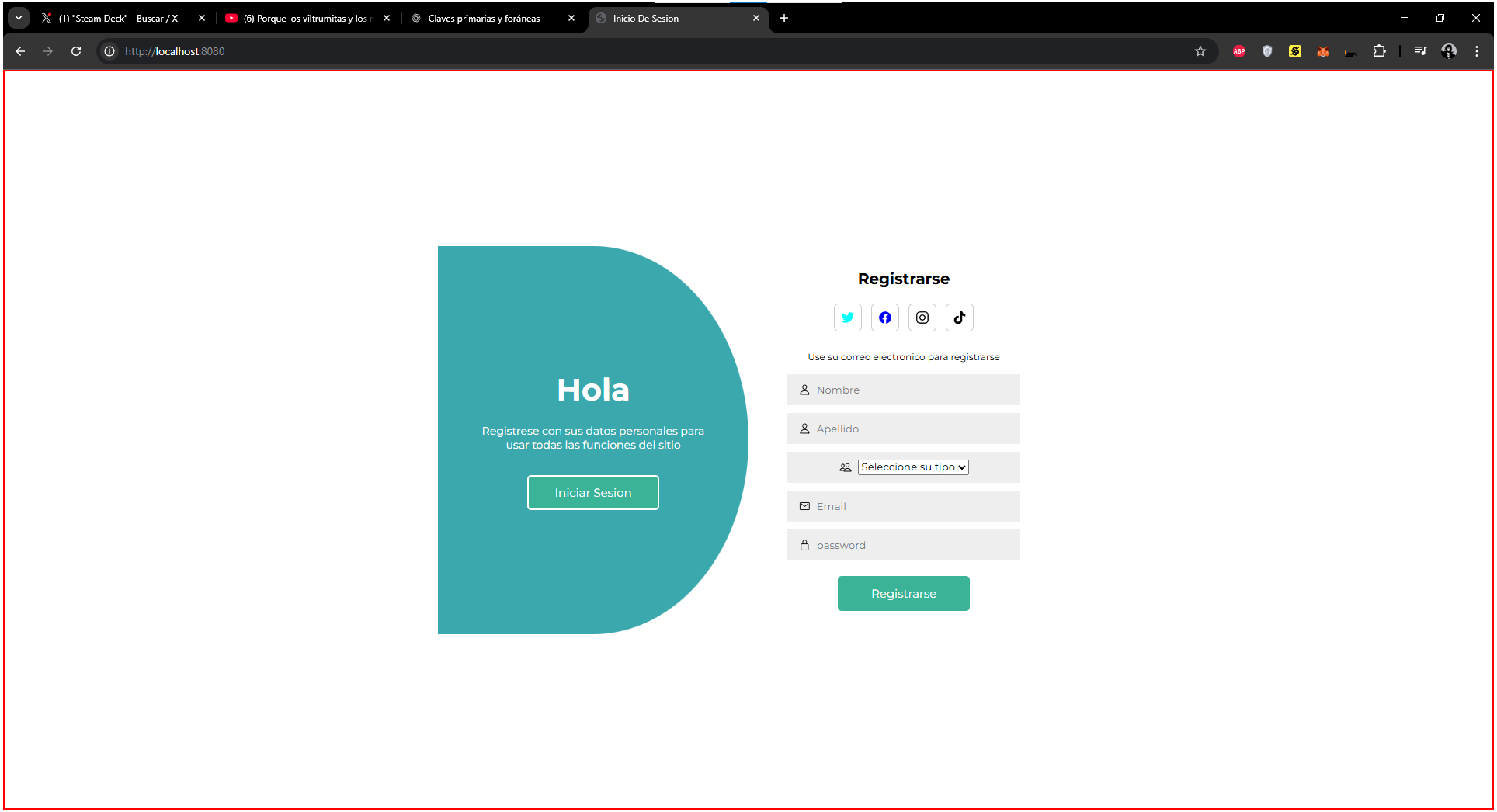
Esta es una de las partes más interesantes del inicializador de Spring Boot, ya que nos permite seleccionar las dependencias necesarias, es decir, las librerías y herramientas que Spring puede proporcionarnos.

Entre las dependencias que seleccionamos se encuentran:

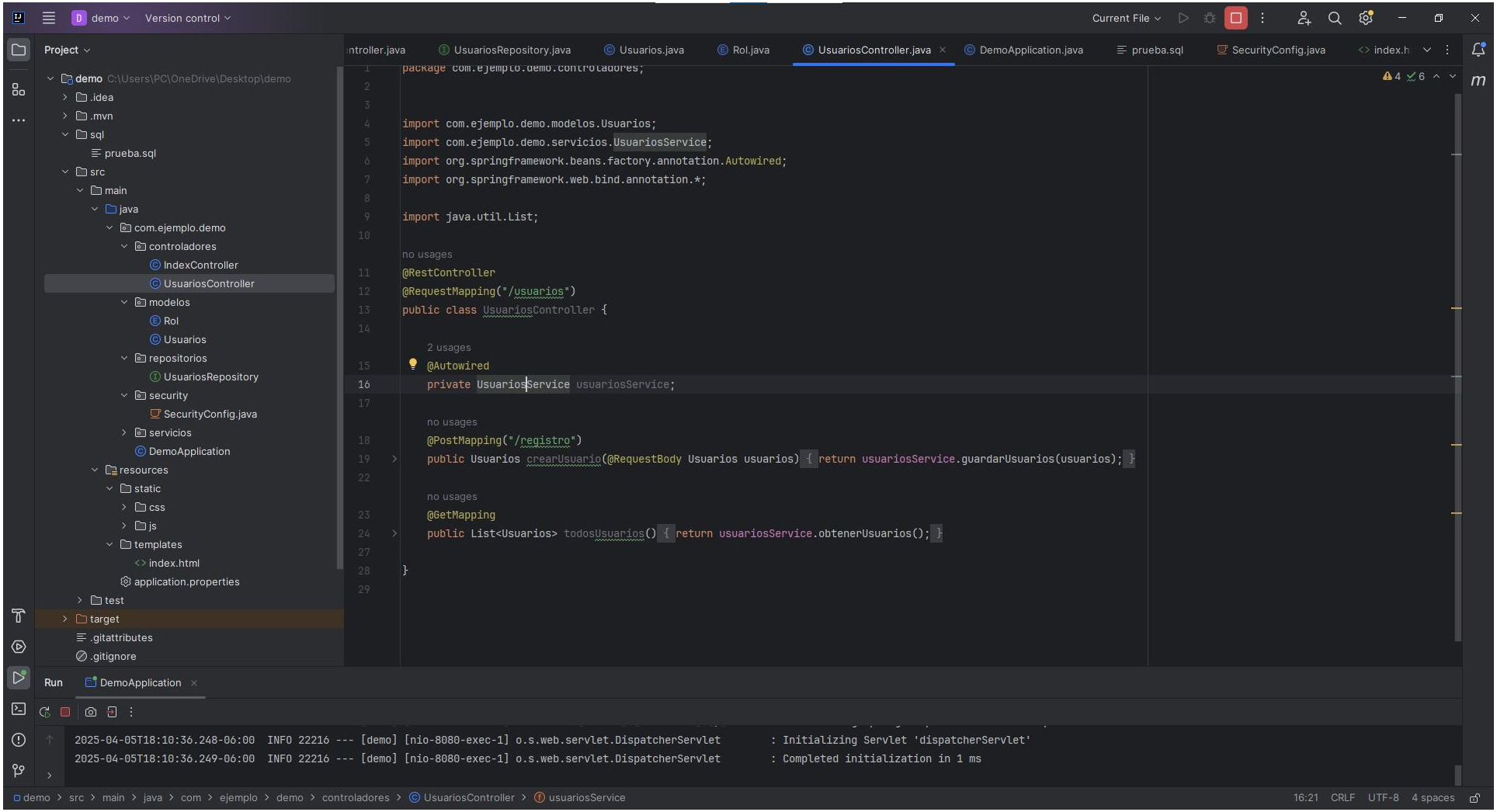
- **MySQL Driver**: nos permite establecer la conexión con una base de datos MySQL.
- **Spring Web**: habilita la creación y ejecución de aplicaciones web. Esta dependencia incluye Apache Tomcat como servidor embebido y permite el desarrollo de APIs RESTful.
- **Spring DevTools**: facilita el desarrollo al reflejar automáticamente los cambios realizados en archivos .java, .html, .js o .css con solo actualizar la página, sin necesidad de reiniciar la aplicación.
- **Thymeleaf**: es un motor de plantillas que permite renderizar páginas HTML de forma dinámica. Facilita la integración de HTML con JavaScript y CSS, y permite incorporar datos antes de mostrar la página al usuario.
- **Spring Data JPA**: permite mapear la base de datos en clases de Java (tablas, vistas, procedimientos almacenados, etc.), lo cual simplifica operaciones como guardar, modificar o eliminar registros sin necesidad de escribir consultas SQL manualmente, ya que proporciona métodos predefinidos para ello.
- **Spring Security**: se encarga de la autenticación y autorización de usuarios. Esta dependencia impide, por ejemplo, que un usuario sin privilegios acceda a áreas reservadas para administradores y viceversa.



Esta es la pagina de inicio que vería tanto un usuario como una autoridad, permite ingresar en la misma pagina de login, ya que spring security es el que se encarga de diferenciar a estas dos entidades.



Esta es la parte del registro, que se encuentra básicamente en la misma vista que la página de login, pero simplemente cambia al presionar un botón, aquí podemos ver cómo se registraría un usuario en la página, aunque esta parte esta un poco desactualizada ya que antes permitíamos que las autoridades se pudieran registra, pero ahora en cambio solo permitimos que los usuarios lo hagan, las autoridades las registraríamos nosotros para mayor seguridad.



Este es un proyecto demo que desarrollamos para comprender el funcionamiento de un proyecto con Spring Boot y las dependencias mencionadas anteriormente. El proyecto está estructurado en dos carpetas principales: com.ejemplo.demo y resources.

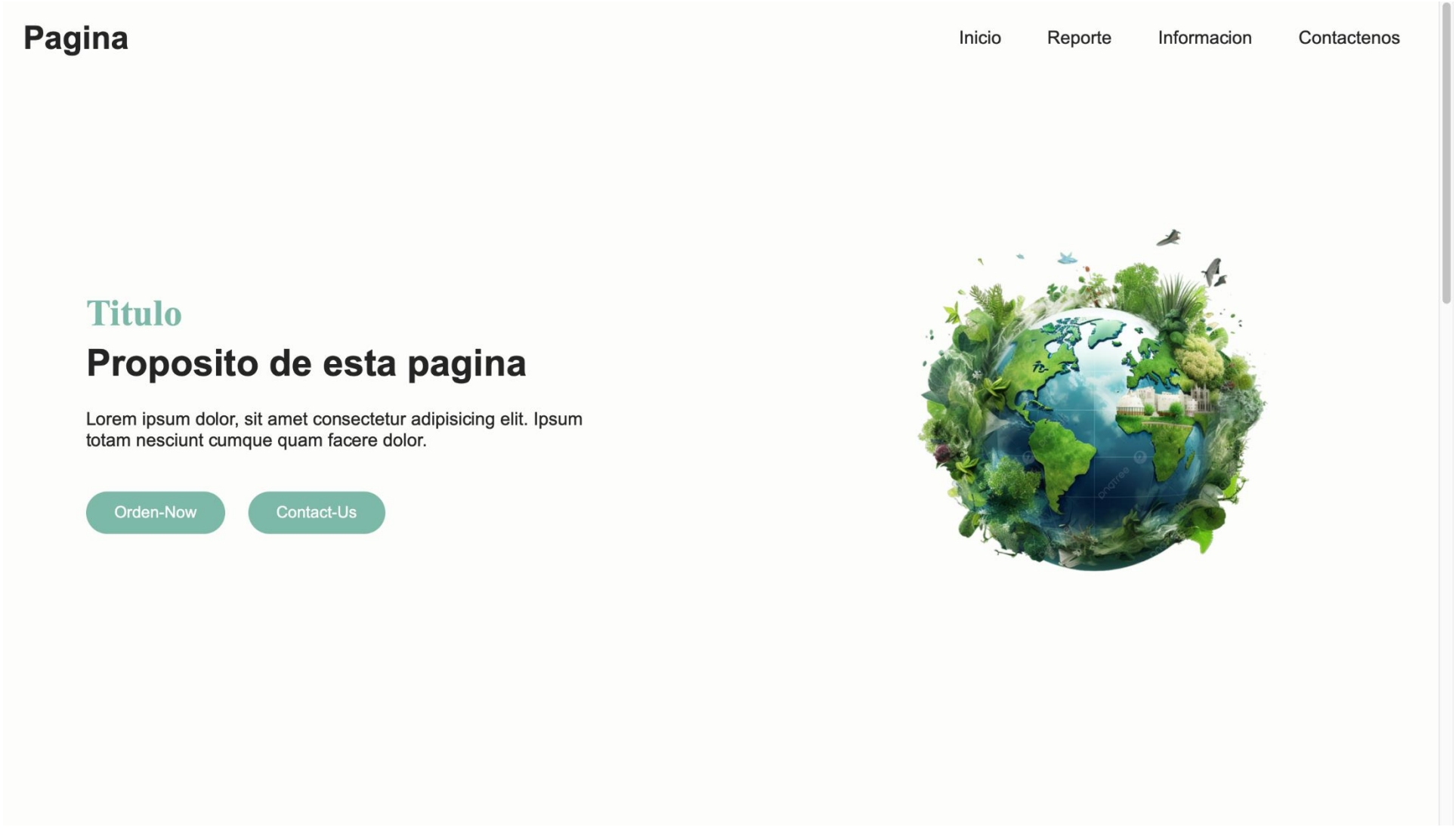
En la carpeta com.ejemplo.demo se encuentra toda la lógica del proyecto. Su estructura está organizada de la siguiente manera:

- **Controladores (controllers)**: contienen la lógica de la API RESTful y gestionan las solicitudes tanto a las vistas (como el login) como a los endpoints utilizados por usuarios y autoridades.

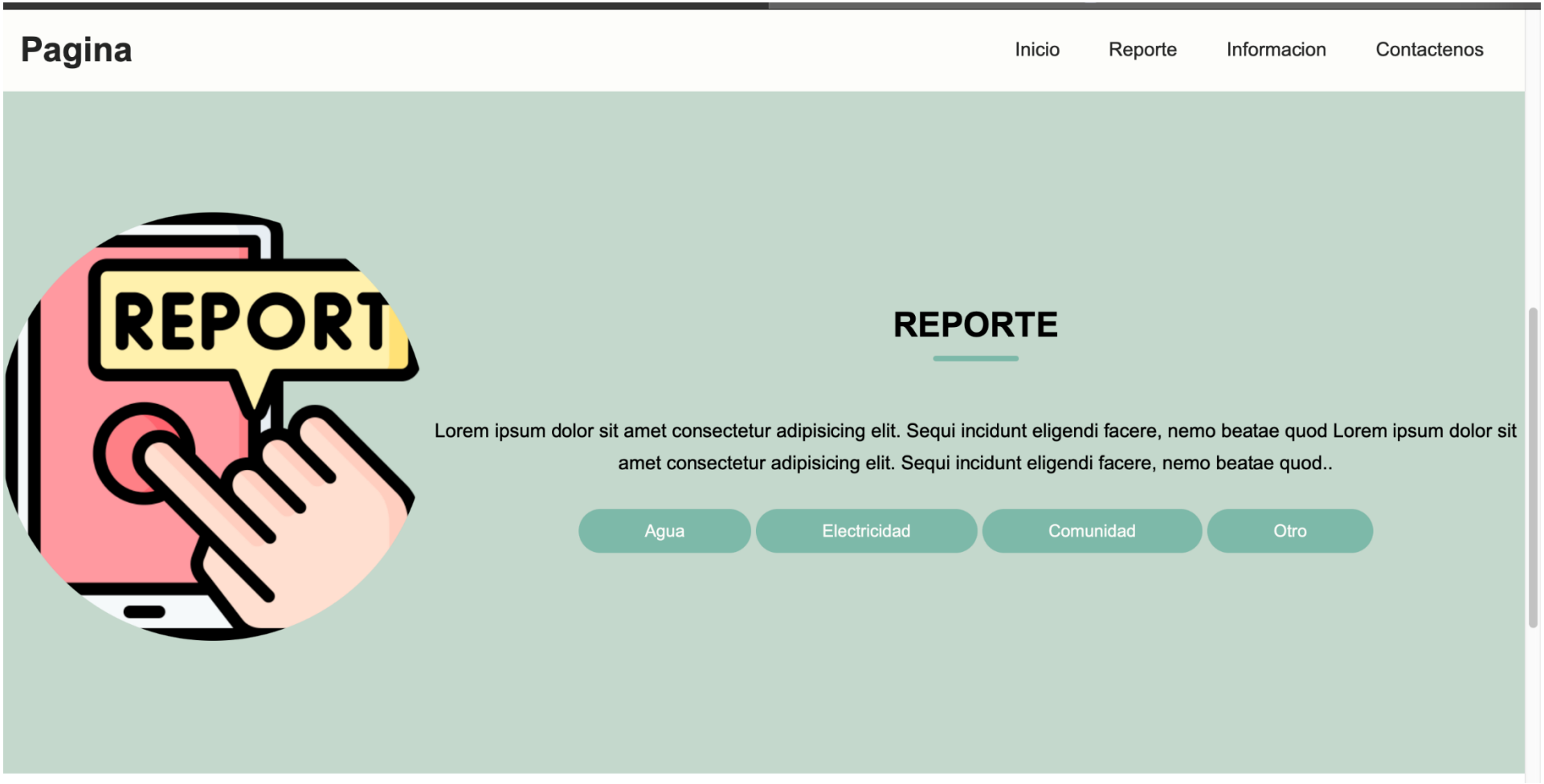
- **Modelos (models)**: representan las tablas de la base de datos, modeladas como clases Java mediante Spring Data JPA.
- **Repositorios (repositories)**: definen las operaciones sobre la base de datos. Incluyen métodos automáticos proporcionados por Spring Data JPA y también permiten la creación de consultas personalizadas.
- **Seguridad (security)**: se encarga de la autenticación y autorización de los usuarios, controlando el acceso según su rol o privilegios.
- **Servicios (services)**: contienen la lógica de negocio de la aplicación, actuando como puente entre los controladores y los repositorios.

Esta organización modular permite encapsular cada componente de la aplicación, facilitando su mantenimiento y permitiendo realizar cambios sin afectar el resto del sistema.

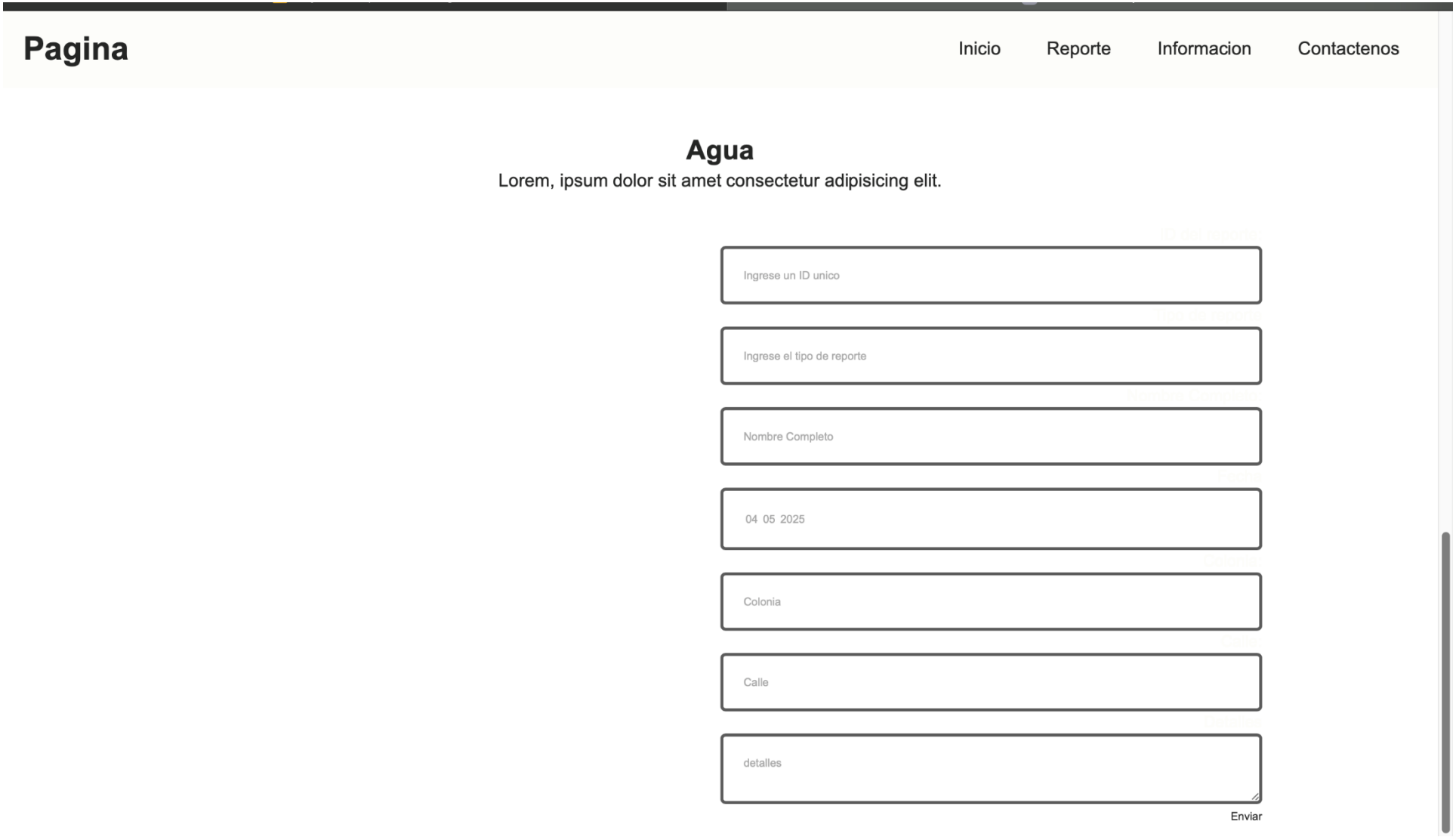
Por otro lado, en la carpeta resources se encuentra la parte visual de la aplicación, es decir, el Frontend. Aquí se almacenan archivos HTML, JavaScript y CSS que componen la interfaz de usuario.



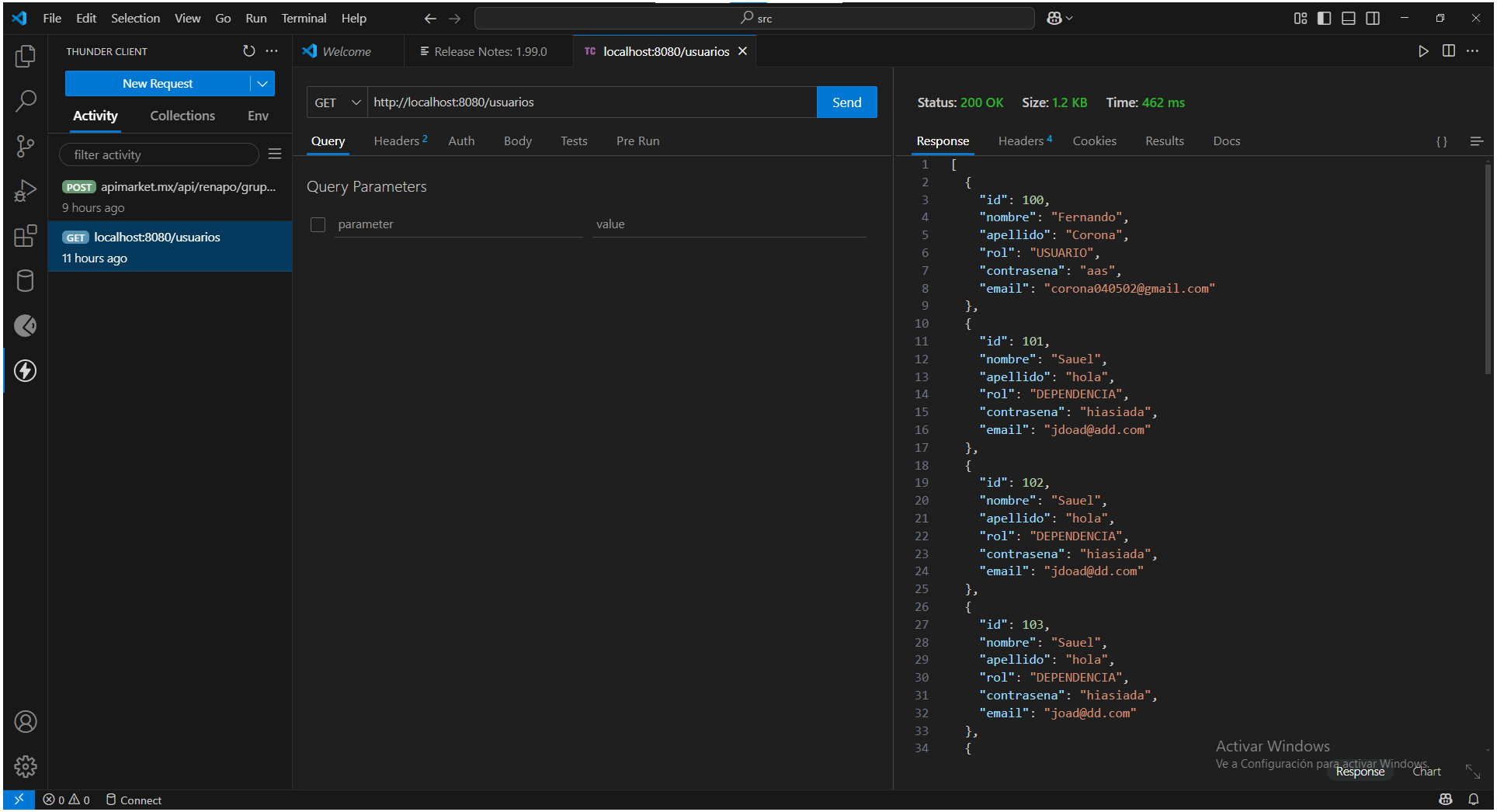
Esto seria lo que vería en un inicio un usuario al hacer login. Tiene texto de ejemplo ya que aun esta en proceso lo que tendría que llevar cada campo de texto



Aquí podemos ver la parte donde el usuario haría el reporte, donde estan 4 botones que, al presionarlos, veríamos un formulario por cada tipo de problema, ya sea de agua, electricidad o etc.

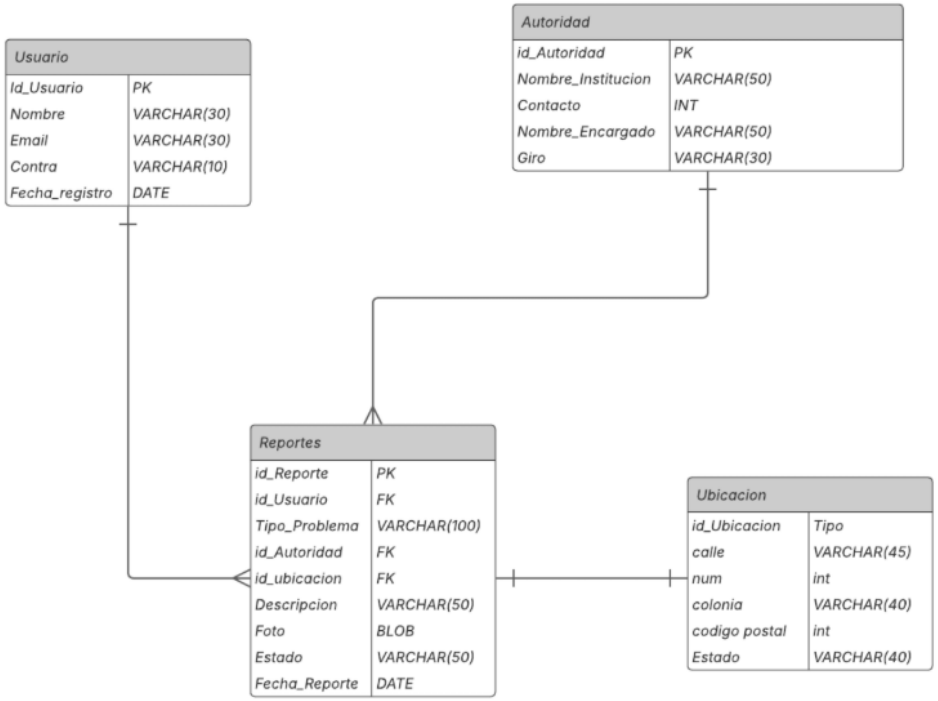


Este sería el formulario para realizar reportes por parte de problemas de agua.

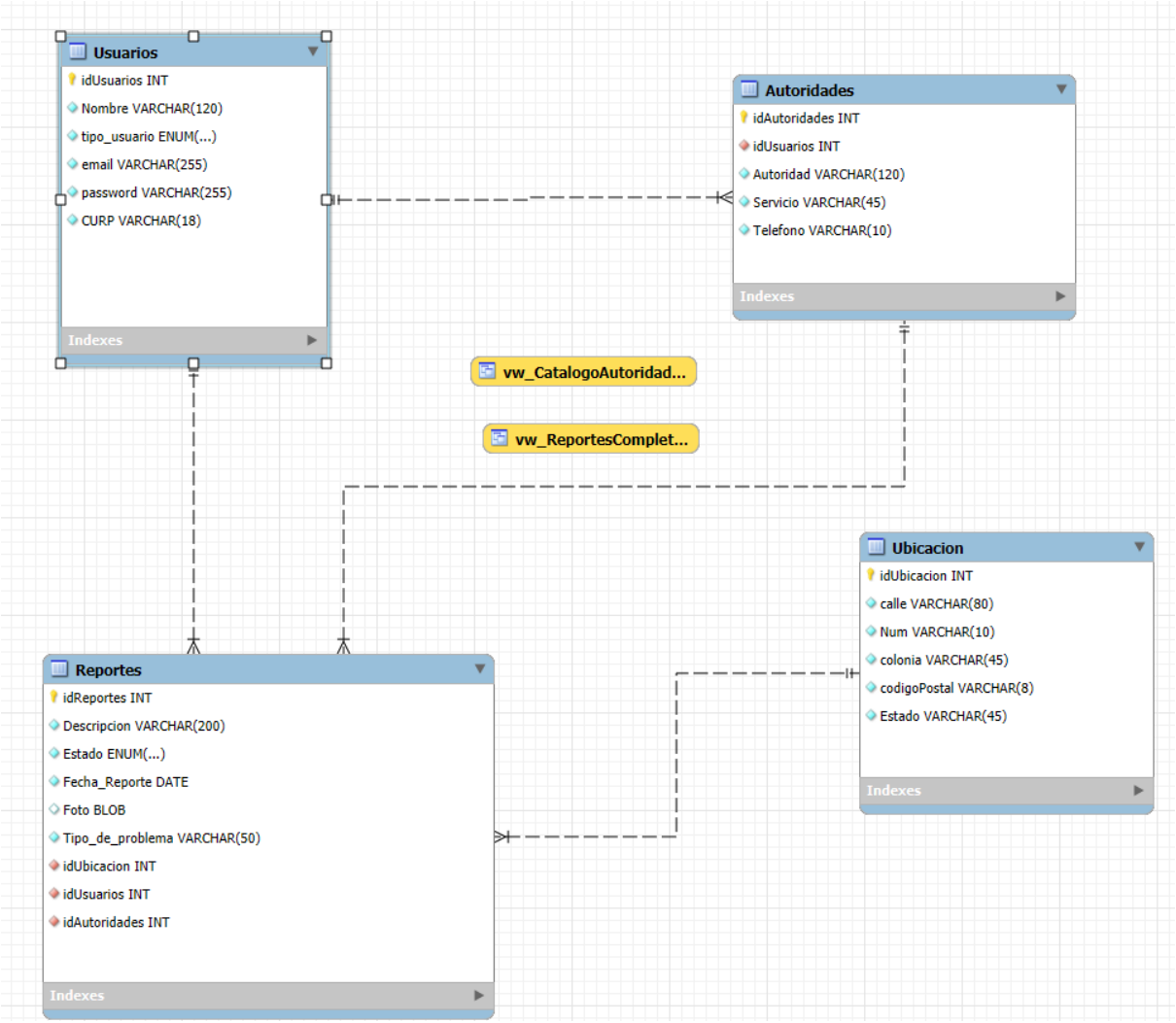


Aquí podemos ver como si estan funcionando los endpoins que se usaron en el demo, ya que teníamos un endpoint en localhost:8080/usuarios, donde al hacer una solicitud ahí, recibimos como respuesta a todos los usuarios.

Esquema de la base de datos



Esta seria la primera versión de nuestra base de datos, como aun no habíamos pensado en que tecnologías usar, no habíamos pensado en que una autoridad es un Usuario, que podria ingresar a la pagina y modificar los estados de un reporte, además de que habia ciertos errores como el contacto era un INT en lugar de varchar



Esta es el esquema de la base de datos más refinado, donde ya incluimos que una autoridad es un usuario que puede ingresar a la aplicación, además que agregamos el campo de curp del usuario, ya que al inicio no habíamos pensado en hacer una verificación de identidad, hasta que vimos que era posible que bots se registraran y realizar reportes y saturar la página, de ahí en mas no hubo grandes cambios en el esquema.

También se agregaron dos vistas, 1 de catálogo de autoridades, donde los usuarios podrías visualizar a cada autoridad, su encargado, teléfono etc. Y otra de todos los reportes, donde se vería el reporte y su ubicación, esto permitiría que los usuarios vean los reportes de otros, mas que nada por si ven que un problema ya fue resuelto, no se repita el mismo problema.

Conclusión

Esta claro que aun falta para terminar nuestro proyecto, pero como refinamos aun mas el esquema de la base de datos, escogimos tecnologías robustas y vimos que funcionan, y hemos avanzado en la parte visual de la aplicación, podemos decir que estamos aun mas cerca de llegar a nuestro objetivo de crear la aplicación.

Las tecnologías seleccionadas, nos van a facilitar la creación de la app, ya que si la hiciéramos desde 0 tendríamos que usar servlets, y pues básicamente Spring abstrae los servlets y nos permite trabajar de manera mas sencilla.

Y el escoger este proyecto es debido a que los espacios urbanos cada día que pasan están más descuidados, esto impacta directamente a la salud de las personas y a la estética de nuestras ciudades o colonias, además de ser una problemática que radica no solo a lo urbano sino también a lo rural, la contaminación es un tema que afecta a todos. Elegimos este proyecto con el objetivo de tomar conciencia que los espacios que habitamos deberíamos de respetarlos y cuidarlos, haciendo un uso correcto de los servicios públicos. ¡¡La responsabilidad es de todos!!