



**ACTUALIZACIÓN DEL MÓDULO PARA EL MANEJO DE LA INSCRIPCIÓN DE
ASPIRANTES DE POSGRADO EN LA UDEA**

Juan David Zuluaga Ruiz

Práctica empresarial como requisito parcial para optar al título de:
Ingeniería de Sistemas

Asesor interno:

Jaime Humberto Fonseca Espinal

Título académico Especialista en ciencias electrónicas e informáticas

Universidad de Antioquia

Facultad de Ingeniería

Medellín, Colombia

Pregrado

2023

Cita	(Zuluaga Ruiz, 2023)
Referencia	Zuluaga Ruiz, J. D. (2023). <i>Actualización del módulo para el manejo de la inscripción de aspirantes de posgrado en la UdeA</i> [Pregrado]. Universidad de Antioquia, Medellín Seleccione ciudad UdeA (A-Z).
Estilo APA 7 (2020)	



Créditos a la división de gestión informática de la Universidad de Antioquia. Dennis Alexandra Marín Marín como Asesora externa de prácticas., Jaime Humberto Fonseca Espinal como asesor interno - Universidad de Antioquia, Equipo de trabajo - división de gestión informática de la Universidad de Antioquia.



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Jesús Francisco Vargas Bonilla.

Jefe departamento: Diego José Luis Botía Valderrama.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

AGRADECIMIENTOS

Agradezco a los docentes de la Universidad de Antioquia con los cuales he tenido la fortuna de compartir y aprender durante este proceso formativo de la carrera de ingeniería de sistemas, no solo por los conocimientos y enseñanzas sino también por todas las experiencias vividas. Agradezco también a Jaime Humberto Fonseca Espinal como asesor interno y a Dennis Alexandra Marín Marín como Asesora externa pues sin su acompañamiento y guía no hubiera sido posible llevar a cabo este proyecto. También agradezco a la división de gestión informática por darme la oportunidad de realizar mis prácticas profesionales en la Universidad de Antioquia y así tal vez retribuir un poco a esta que me lo ha brindado todo. Por ultimo me gustaría agradecer a la división de gestión informática y en especial al equipo de desarrollo web por toda su colaboración y enseñanzas brindadas durante este proceso.

Tabla de contenido

1	Introducción	9
2	Planteamiento Del Problema	10
3	Justificación	11
4	Objetivos	12
	4.1	12
	4.2	12
5	Marco Teórico	13
6	Metodología	16
7	Resultados Y Análisis	18
8	Pruebas	27
9	Documentación, Implementación, Mantenimiento Y Soportes	28
10	Resultados Finales	30
11	Dificultades	32
12	Conclusiones	33
13	Referencias Bibliográficas	34

Lista de figuras

Figura 1 Flujo del proceso de inscripción planteado en el sistema reDoc.....	21
Figura 2 Flujo del proceso de inscripción con archivos en servidor (sin integración a reDoc)	22
Figura 3 Ejemplo de un llamado SQL y mapeo de entidad en JDBC.....	22
Figura 4 Ejemplo de un llamado SQL por medio del ORM JPA.....	22
Figura 5 Formulario de inscripción.....	23
Figura 6 Sección de datos generales del formulario de inscripción	23
Figura 7 Prueba de envío de correo en inscripción exitosa.....	24
Figura 8 Lista de requisitos en módulo de admisiones.....	24
Figura 9 Edición de resolución en módulo de admisiones.....	25
Figura 10 Búsqueda de programa por código o nombre en módulo de admisiones	25
Figura 11 Lista de programas con revisor asignado nombre en módulo de admisiones	26

Siglas, Acrónimos Y Abreviaturas

API	Application Programming Interface
REST	Representational State Transfer
JDBC	Java Database Connectivity
SQL	Structured Query Language
JPA	Java Persistence API
ORM	Object Relational Mapping
JSF	JavaServer Faces
JPQL	Java Persistence Query Language
UdeA	Universidad de Antioquia

Resumen

El aplicativo de inscripción posgrado de la Universidad de Antioquia se encarga tanto del proceso de inscripción de los aspirantes a los diferentes programas de posgrado ofrecidos por la universidad como también del manejo y revisión de datos de los inscritos por parte de la unidad administrativa. Este aplicativo se encuentra actualmente desarrollado en tecnología java con icefaces en una arquitectura monolítica y desplegada en el antiguo servidor de aplicaciones de la Universidad de Antioquia. En tiempos recientes el antiguo servidor de aplicaciones de la Universidad de Antioquia ha demostrado un detrimento sustancial en su funcionamiento, por lo cual se han ido migrando las aplicaciones hacia un nuevo servidor de aplicaciones. La decisión de la migración y actualización del módulo de inscripción posgrado se toma en medio de la adopción de una arquitectura más cercana a API RESTfull que se está llevando en el área de gestión informática, teniendo en cuenta esto la actualización del aplicativo tendrá efecto tanto en la modernización del módulo pero al mismo tiempo también aportará a aumentar la escalabilidad y disminuir el acoplamiento.

En el desarrollo también se decidió dividir el módulo en 2 aplicativos, uno encargado de la inscripción de aspirantes desarrollado en las tecnologías anteriormente mencionada y la persistencia en java se manejó mediante el motor JDBC “plano” que se utiliza actualmente en los aplicativos desarrollados en la división, mientras que el otro modulo es el encargado de la gestión y revisión de datos de los estudiantes y con el cual se decidió implementar el uso de la interfaz JPA para la persistencia de datos que ayudaría como prueba de concepto para su posible adopción de forma más generalizada en otros proyectos de la unidad.

Abstract

The postgraduate enrollment application of the University of Antioquia is in charge of the enrollment process of applicants to the different postgraduate programs offered by the university as well as the management and review by the administrative unit of data from those enrolled. This application is currently developed in java with ice faces in a monolithic architecture and deployed in the old application server of the University of Antioquia. In recent times the old application server of the University of Antioquia has shown a substantial detriment in its performance so the applications have been gradually migrated to a new application server. The decision to migrate and update the postgraduate enrollment module is taken in the midst of the adoption of an architecture closer to RESTfull API that is being carried out by the IT management area, taking this into account the application update will not only have an effect on the modernization of the module but will also contribute to increase scalability and reduce coupling. For this development it was also decided to divide the module into 2 applications, one in charge of the registration of applicants developed in the aforementioned technologies with the persistence in java handled by the JDBC "flat" engine that is currently used in the applications developed by the division, while the other is responsible for the management and review of student data and with which it was decided to implement the use of the JPA interface for the persistence of data that would help as a proof of concept for its possible adoption in a more generalized way in other projects of the unit.

1 Introducción

Dentro de la Universidad de Antioquia se encuentra la división de gestión informática, cuyo objetivo es proveer servicios de tecnologías de la información y la comunicación para el apoyo de los procesos institucionales. Adicionalmente, en esta se encuentra la unidad de desarrollo web que se encarga del desarrollo, actualización y soporte de los aplicativos y módulos web utilizados por los empleados y alumnos de la universidad.

Una de las funciones de la unidad de desarrollo web es el de proveer soporte y actualizaciones a los aplicativos y módulos web que se utilizan en los diferentes procesos de la Universidad, uno de estos aplicativos es el que lleva el proceso de la inscripción de los estudiantes de posgrado, el cual actualmente se encuentra desarrollado en la tecnología java por medio de jsf. Se plantea la necesidad de la actualización de este módulo migrando hacia el modelo API REST e interfaz de usuario en Angular. Para esto es necesario realizar un análisis del aplicativo actual para comprender su composición y funcionamiento y así plantear los posibles cambios necesarios para llevar a cabo la migración satisfactoriamente.

Finalmente, cabe destacar que, como parte del desarrollo de esta práctica académica, también se llevó a cabo de forma paralela al desarrollo del módulo de posgrado las actividades de soporte y mantenimiento propias del equipo de desarrollo web.

2 Planteamiento Del Problema

El antiguo servidor de aplicaciones de la Universidad de Antioquia (Sikuani) ha venido demostrado un detrimento significativo de su performance, en especial en los últimos años la estabilidad de las aplicaciones desplegadas en este mismo se ha visto afectada notablemente generando muchos inconvenientes en el normal desarrollo de un amplio rango de las actividades administrativas de la Universidad, lo cual finalmente ha venido afectando a un gran porcentaje de la comunidad universitaria. Una de las aplicaciones que aún se encuentra desplegada en este servidor y por tanto se ve afectada es el módulo encargado del proceso completo de inscripción de aspirantes a los programas de posgrado de la Universidad de Antioquia.

Por otra parte, el proceso de migración de las aplicaciones hacia un nuevo servidor ha dado la oportunidad de también migrar las mismas hacia una nueva arquitectura con una nueva metodología, por lo cual en el equipo de desarrollo web de la división de gestión informática se ha venido adoptando la metodología agile, específicamente una metodología cercana a SCRUM y un desarrollo cercano a la arquitectura API REST.

3 Justificación

La migración del aplicativo de inscripción para aspirantes a los programas de posgrado se sustenta en la necesidad de que esta esté siempre disponible para los constantes procesos de admisión que realiza la universidad, disponibilidad que se podría ver en riesgo debido a la inestabilidad que presenta el servidor Sikuni donde esta se encuentra actualmente desplegada. Debido a esto la división de gestión informática ha venido implementando un plan de migración de aplicaciones hacia un servidor nuevo lo cual ha demostrado una mejora en la disponibilidad y respuesta de las aplicaciones con las cuales se ha realizado dicho proceso de migración. Más adelante se realizó otro plan de migración de aplicaciones por parte de la división de gestión informática el cual consiste en el paso hacia el uso de integración continua y el uso de un servidor de aplicaciones en la nube.

Por otra parte, la decisión de desarrollar la aplicación de nuevo migrando hacia nuevas tecnologías se da debido a que la aplicación actual fue desarrollada hace bastantes años, la arquitectura monolítica en la cual esta se encuentra desarrollada así como las tecnologías usadas hacen bastante compleja la mantenibilidad del módulo, así como también limita en gran medida la escalabilidad. La decisión de usar una metodología cercana a API REST ayuda enormemente a reducir el acoplamiento en la aplicación y proveer escalabilidad para la misma.

Por último la decisión de dividir el aplicativo en 2 módulos (inscripción de aspirantes y manejo de admisiones) ayuda en gran medida en la modularidad lo cual se ve traducido en una mantenibilidad mayor y da la oportunidad de probar el uso de Jakarta Persistence API lo cual podría ayudar a la adopción de esta tecnología lo para el manejo de entidades en código y su gestión en bases de datos lo cual finalmente se vería reflejado en mayor agilidad a la hora de desarrollar futuras aplicaciones.

4 Objetivos

4.1 Objetivo general:

- Implementar nuevamente el módulo para el manejo de la inscripción de aspirantes de posgrado en la UdeA hacia un modelo API RESTful.

4.2 Objetivos específicos:

- Analizar el aplicativo existente y su funcionamiento.
- Planear y analizar la actualización y plantear la metodología para la migración del aplicativo hacia las nuevas tecnologías.
- Analizar las modificaciones necesarias para llevar a cabo la migración.
- Desarrollar e implementar los módulos backend por medio de Java spring boot.
- Desarrollar e implementar los front-end por medio del framework Angular.
- Realizar las pruebas necesarias para asegurar el funcionamiento del módulo.
- Documentar el trabajo realizado.

5 Marco Teórico

La migración se realizará en 2 partes, la primera el back-end el cual consta de APIs REST desarrolladas utilizando el framework Java Spring boot y herramientas como javamelody, LogBack, Oracle jdbc connector y otras propias de la Universidad. La segunda el front-end el cual se desarrollará a través del framework Angular.

API: Una API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de las aplicaciones. Las APIs permiten la comunicación entre productos y servicios sin la necesidad de que estos conozcan la implementación de unos y otros.

REST: El término Transferencia de Estado Representacional (Representational State Transfer) o REST, fue introducido y definido en el año 2000 por Roy Fielding es una arquitectura de desarrollo web que se puede utilizar en cualquier cliente HTTP basada en los métodos GET, POST, PUT, DELETE y de gran popularidad debido a la gran escalabilidad y simplicidad que ofrece frente a sus contrapartes. Sus características son:

- Es un protocolo **sin estado**, debido a que no se guarda la información en el servidor. Es decir, toda la información será enviada por el cliente en cada mensaje HTTP, consiguiendo un ahorro en variables de sesión y almacenamiento interno del servidor.
- Presenta un conjunto de operaciones bien definidas, siendo las más importantes *GET*, *POST*, *PUT* y *DELETE*, que se emplea en todos los recursos.
- Utiliza **URIs** únicas siguiendo una sintaxis universal. Estas URIs deben seguir unas pautas que veremos más adelante.
- Emplea *hipermedios* para representar la información, que suelen ser **HTML**, **XML** o **JSON**.

JAVA: Java es un lenguaje de programación y una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems. Java es un lenguaje de propósito general y uno de los más populares actualmente y destaca por su versatilidad y soporte prácticamente universal.

JSF: La tecnología JavaServer Faces es un framework de interfaz de componentes de usuarios del lado del servidor para las aplicaciones web basadas en la tecnología Java. Los principales componentes de la tecnología JSF son:

- Una API para representar componentes de Interfaz de Usuario (UI) y gestionar su estado.
- Manejar eventos, validar en el servidor y conversión de datos.
- Definir la navegación de páginas.
- Soporte de internacionalización y accesibilidad.
- Dos librerías de etiquetas JSP personalizadas para expresar componentes en una página JSP y enlazar los componentes a objetos del servidor.

JPQL: JPQL es un lenguaje de consultas que se basa en comandos sql pero con una variación ya que se usa la caché de la persistencia usado para hacer consultas contra las entidades almacenadas en una base de datos relacional. Está inspirado en gran medida por SQL, y sus consultas se asemejan a las consultas SQL en la sintaxis, pero opera con objetos entidad y también notar que hemos hecho trampa si intentamos copiar esta parte de aquí mismo de JPA en lugar de hacerlo directamente con las tablas de la base de datos.

SPRING: JAVA SPRING fue escrito inicialmente por Rod Johnson como una plataforma java de código abierto y se ha convertido en el framework más popular de JAVA empresarial, su finalidad es estandarizar, agilizar, manejar y resolver los problemas que puedan ir surgiendo en el trayecto de la programación ya que ofrece como elemento clave el soporte de infraestructura a nivel de aplicación, brindando un completo modelo tanto para la configuración como para la programación de aplicaciones empresariales desarrolladas bajo Java, sin discriminación en cuanto al

despliegue de la plataforma. SPRING framework es mayormente conocido por la inyección de dependencias, pero otras de sus características ofrecidas son [9]:

- **Core container:** proporciona inyección de dependencias e inversión de control.
- **Web:** nos permite crear controladores Web, tanto de vistas MVC como aplicaciones REST.
- **Acceso a datos:** abstracciones sobre JDBC, [ORMs](#) como Hibernate, sistemas OXM (*Object XML Mappers*), JSM y transacciones.
- **Programación orientada a Aspectos (AOP):** ofrece el soporte para [aspectos](#).
- **Instrumentación:** proporciona soporte para la instrumentación de clases.
- **Pruebas de código:** contiene un *framework* de *testing*, con soporte para JUnit y TestNG y todo lo necesario para probar los mecanismos de Spring.

SPRING BOOT: Spring Boot simplifica al máximo el proceso de configuración inicial y preparación de las aplicaciones para producción gracias a sus dos principales mecanismos.

- **Contenedor de aplicaciones integrado:** Spring Boot permite compilar nuestras aplicaciones Web como un archivo .jar que podemos ejecutar como una aplicación Java normal (como alternativa a un archivo .war, que desplegaríamos en un servidor de aplicaciones como Tomcat).
- **Starters:** Spring Boot nos proporciona una serie de dependencias, llamadas starters, que podemos añadir a nuestro proyecto dependiendo de lo que necesitemos: crear un controlador REST, acceder a una base de datos usando JDBC, conectar con una cola de mensajes Apache ActiveMQ, etc.

6 Metodología

El equipo de desarrollo web implementa una metodología de desarrollo la cual toma varias características de SCRUM. SCRUM es una metodología de desarrollo ágil que se basa en el aprendizaje continuo y la adaptación a factores fluctuantes, esto lo hace por medio entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto.

La metodología a usar toma prestado el core de SCRUM, en la recepción inicial del aplicativo se analiza el proyecto y se realiza una planificación básica del ciclo de desarrollo y de la estructura de la aplicación basada en la aplicación existente a migrar para realizar el proceso de desarrollo en ciclos semanales en los cuales se realizan reuniones de control y retroalimentación entre el desarrollador y el arquitecto encargado además de reuniones constantes de retroalimentación y planificación de cara a los cambios necesarios en la base de datos y/o la aplicación a desarrollar frente a la ya existente.

Actividades a realizar:

- 1) Capacitación inicial:
 - a) Conocer acerca de los procesos y arquitectura usada en las aplicaciones web de la división de gestión informática.
 - b) Estudiar la documentación inicial, manuales y lineamientos de desarrollo de la división de gestión informática.
 - c) Instalación y configuración del entorno de desarrollo de software, así como de las librerías necesarias.
- 2) Recibir aplicación a migrar, reunión con analista de capa web y la arquitecta de software para explicar el contexto y funcionalidad de la aplicación antigua.
- 3) Analizar la necesidad de nuevos procedimientos o funciones en la base de datos, esto implica reunión y retroalimentación constante con el analista de capa web,

el arquitecto de software y el dueño del esquema de la base de datos según se vaya presentando la necesidad.

- 4) Analizar la aplicación existente, proceso constante de análisis del código y funcionamiento de la aplicación a migrar.
- 5) Desarrollar el back-end: se desarrolla la lógica del negocio y la persistencia de datos mediante java spring boot, esto conlleva también analizar la posible división en diferentes módulos back-end siguiendo las premisas de API RESTful sin llegar al punto de saturar el aplicativo con demasiadas divisiones.
- 6) Desarrollar el front-end: crear las páginas y módulos necesarios en Angular.
- 7) Realizar el despliegue del aplicativo en el ambiente de pruebas para así garantizar la funcionalidad, así como también detectar y documentar posibles bugs, problemas y errores.
- 8) Elaborar la documentación necesaria.
- 9) Llevar a cabo de manera paralela las labores de soporte propias de la división de gestión informática según se vayan presentando.
- 10) Realizar de manera paralela el análisis y retroalimentación propias de la migración del aplicativo.
- 11) Asistir a las reuniones de seguimiento semanales.

7 Resultados Y Análisis

En pos de entregar una mayor claridad a los resultados se abordaron estos desde cada una de las fases o puntos propuestos en la metodología descrita anteriormente.

En la primera fase se realizó la capacitación inicial, familiarizándose con el equipo de trabajo y la división de gestión informática en general además de cómo se lleva proceso de desarrollo con los manuales institucionales, se instala y configura los entornos de desarrollo siguiendo dichos manuales.

Luego se realizaron las reuniones con el arquitecto encargado de la aplicación para la contextualización y entrega de la aplicación, se explicó el funcionamiento y los puntos clave de la inscripción de la aspirante, las consultas necesarias en la base de datos y se da un overview de algunos de los posibles cambios iniciales que se plantean a la aplicación al momento incluyendo el deseo de cambiar el almacenamiento y gestión de los archivos de los inscritos hacia el sistema unificado ya establecido por la universidad llamado ReDoc. También se contextualiza sobre un desarrollo en paralelo el cual lleva algunos años en desarrollo y que implementaría la sección de creación de resoluciones necesarias para la publicación de las ofertas del proceso de inscripción de posgrado y de la cual dependería el alcance del módulo de Admisiones encargado de la revisión y gestión de los inscritos a los programas de posgrado.

Al momento de iniciar el desarrollo se analizó las necesidades con la arquitecta y se llegó a la conclusión que la mejor aproximación sería la de desarrollar primero el módulo de inscripción de aspirantes, desarrollando ambos back-end y front-end paralelamente. Inicialmente la estrategia era replicar la funcionalidad de la aplicación antigua, exceptuando el cambio al sistema ReDoc para la gestión de documentos. Pensando en acoplarse a este cambio el primer desarrollo fue el del servicio web wsRedoc el cual era necesario para gestionar los datos de los requisitos asociados a dichos documentos de las tablas de base de datos usadas en el antiguo programa y acoplarlas al sistema ReDoc.

El desarrollo implicaba el análisis del código de la aplicación antigua e implementación en las nuevas tecnologías. Inicialmente en la parte del back-end desarrollada en java los cambios funcionales fueron mínimos, más allá de los necesarios inherentes a la actualización (la aplicación es del 2002) lo cual básicamente implicaba una mejor implementación del modelo vista-controlador y un manejo un poco diferente en el mapeo de las entidades. Por otro lado el cambio en el front-end fue bastante drástico, la diferencia entre el funcionamiento anterior por medio de servlets y páginas jsf en java comparado con la nueva implementación en Angular es demasiado grande, por lo cual migrar de la primera implementación la cual combinaba lógica front y back en el mismo código hacia la nueva aproximación API REST requirió de un análisis más exhaustivo para separar adecuadamente cuáles partes del proceso se debían implementar con java en el back y cuales en TypeScript en el front Angular.

Como se acordó durante la etapa de planeación del proyecto, el manejo de la persistencia del módulo de la inscripción del aspirante se realizó por medio del motor JDBC y el flujo del proceso de inscripción no cambia mucho, en un principio solo había unas cuantas consultas extra relacionadas con como cruzar información con el sistema de reDoc para el manejo de archivos, pero este manejo se realiza por medio del web service para ligar los datos de la aplicación de inscripción posgrados a la aplicación que se desarrollaría en paralelos para los aspirantes y su manejo de datos en reDoc.

También durante el proceso de desarrollo se realizaron reuniones, no solo con el equipo de trabajo, sino que también con la dependencia de admisiones la cual sería el cliente. Durante estas reuniones la dependencia pidió que se tomará el desarrollo teniendo en cuenta la aplicación que se estaba desarrollando paralelamente para el manejo de resoluciones y requisitos, esto también tendría inferencia en el manejo de los procedimientos y tablas usadas para el proceso de los archivos ya que estos están ligados a los requisitos. Sin embargo el desarrollo de este aplicativo paralelo lleva un retraso bastante considerable y en reuniones con la arquitecto y los encargados de dicho desarrollo se llegó a la conclusión de realizar todo el desarrollo sin tener en cuenta dicha aplicación, esto afectaba el módulo de inscripción pero sobre todo el

administrativo y teniendo en cuenta el tiempo y el impacto en el proceso también se decidió hacer el manejo de archivos directamente y dejar la integración a reDoc como un proyecto a futuro.

La implementación realizada siguió el flujo del proceso de la antigua aplicación, esto es, primero el usuario llega re direccionado luego de realizar el pago en la página de la universidad y se valida de manera transparente para el usuario algunos datos, luego se pide que ratifique su documento de identidad y referencia de pago para así, luego de más validaciones de datos en la base de datos, se redirija a la parte del proceso que le corresponde, esta puede ser elegir un programa de la resolución a la cual pagó la inscripción si todavía no lo ha hecho, un mensaje si la resolución por algún motivo (generalmente los tiempos de inscripción pasaron) no tiene programas activos, la credencial de inscripción si ya ha realizado esta y quiere verificar sus datos o el formulario de inscripción con o sin datos precargados dependiendo de si va a rectificar estos o si va a inscribirse por primera vez. Sin embargo, durante las conversaciones con el ingeniero encargado del desarrollo anterior mientras este realizaba pruebas al nuevo aplicativo se decidió eliminar la rectificación del número de documento y referencia de pago del aspirante ya que resultaba redundante.

En la implementación del módulo administrativo para la gestión de los aspirantes inscritos no hubo muchos cambios en cómo se realizan los procesos frente a la versión anterior, los cambios planteados por cómo se manejan los documentos subidos por los inscritos con la integración a la plataforma reDoc se descartaron de momento debido a restricciones de tiempo y disponibilidad del desarrollador de reDoc para los módulos necesarios en este sistema, sin embargo con el webservice desarrollado y unos cuantos cambios en la lógica del back se podría implementar en un futuro.

El cambio más importante en este módulo es la utilización de JPA para la persistencia de datos, esto trae varios retos debido a cómo están implementadas las entidades en el aplicativo anterior y las tablas en la base de datos. Se tuvo que plantear en cada paso cuando era pertinente hacer uso de los beneficios de JPA para la facilitación del mapeo de entidades utilizando las anotaciones, de la interfaz JpaRepository con sus

métodos query ya establecidos y con el mapeo automático de las entidades contra las tablas de las bases de datos, cuando se podía hacer uso de su lenguaje JPQL y cuando definitivamente se tenía que hacer el trabajo extra de realizar las consultas con SQL nativo conllevando el trabajo extra de definir el mapeo de la entidad.

Figura 2 Flujo del proceso de inscripción con archivos en servidor (sin integración a reDoc)

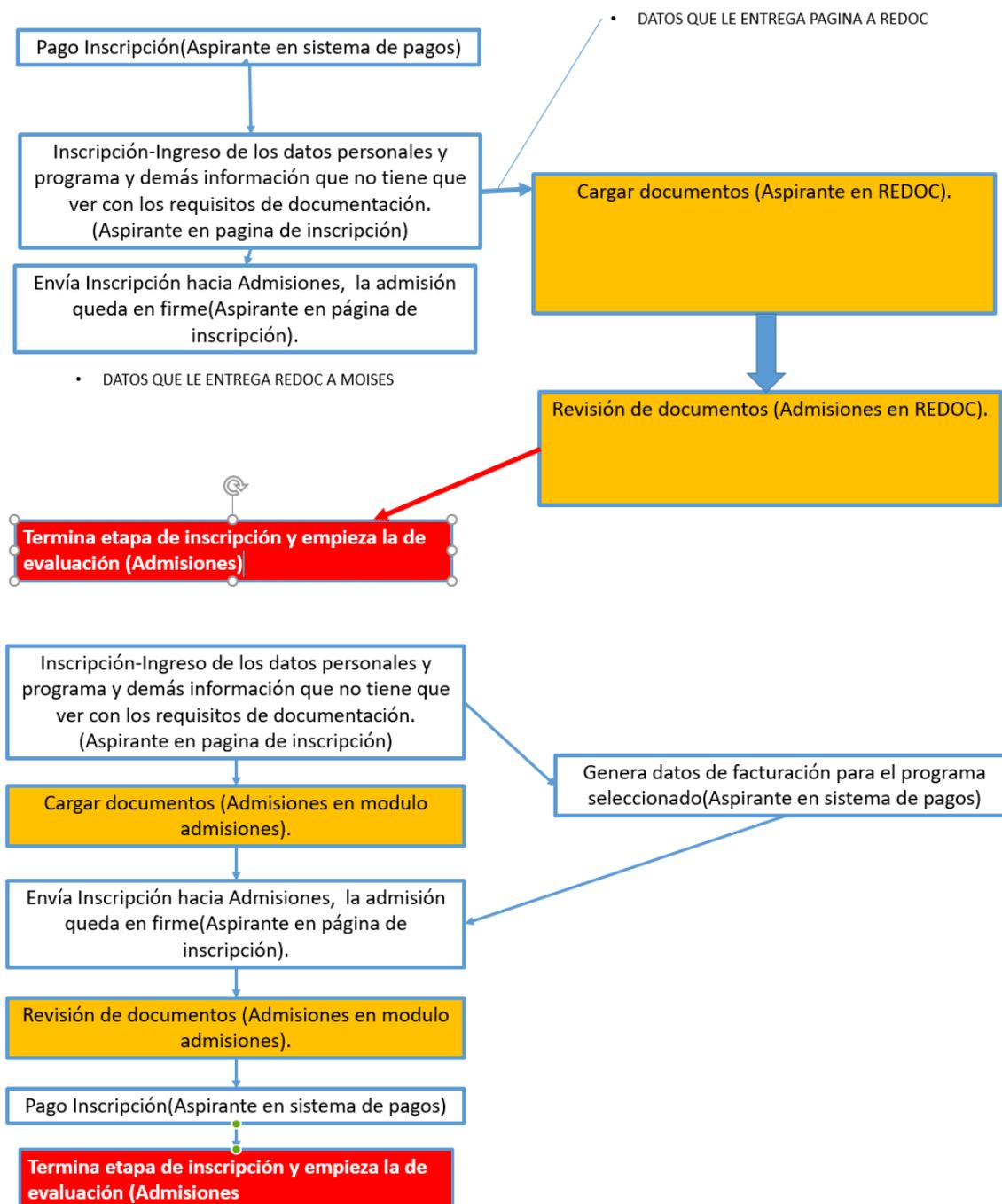


Figura 3 Ejemplo de un llamado SQL y mapeo de entidad en JDBC

```
res = jdbcTemplate.queryForObject(sql, params, (rs, rowNum) ->{
    Entity dato = new Entity();
    dato.setDato(rs.getInt("dato"));
    dato.setDato2(rs.getString("dato2"));
    return dato;
});
```

Figura 4 Ejemplo de un llamado SQL por medio del ORM JPA

```
@Query(nativeQuery = true, name = "getAdmisionesPrograma")
public List<IdentifAdmision> getAdmisionesPrograma(@Param("programa") Long programa);
```

Figura 5 Formulario de inscripción

Posgrados - Inscripción

Eventos	Ten en cuenta las fechas y eventos del calendario para la admisión en el programa	▼
Datos Generales	Ingresar aquí la información personal	▼
Información académica	Ingresar aquí la información institucional y académica de pregrado	▼
Solicitud exención	Solicita una exención de matrícula en caso de cumplir con los requisitos	▼
Cómo se enteró de la convocatoria?	Indicanos como te enteraste de la convocatoria seleccionando la opción pertinente	▼
Línea de investigación	Selecciona la línea de investigación de tu interés	▼
Documentos y requisitos	Verifica los requisitos para tu inscripción y sube aquí los archivos necesarios	▼

Acepto las condiciones establecidas en la [Política de Tratamiento de Información y Protección de Datos Personales de la Universidad de Antioquia](#)

La información ingresada en este formulario se entiende es suministrada bajo la gravedad de juramento y será de exclusiva responsabilidad de quién diligencia el registro y/o presenta las pruebas de admisión. Suministrar información falsa podrá acarrear consecuencias tanto desde el punto de vista administrativo (anulación de la inscripción, la admisión o la matrícula), como penal.

Enviar
Regresar

Figura 6 Sección de datos generales del formulario de inscripción

Datos Generales Ingresar aquí la información personal

Tipo de documento: *
CEDULA DE CIUDADANIA

Documento: *
97254023

Primer apellido * campo requerido

Segundo apellido * campo requerido

Nombres * campo requerido

Segunda lengua

Selecciona tu fecha de nacimiento *

El aspirante debe tener entre 18 y 100 años de edad.

Sexo:

Masculino Femenino campo requerido

Correo electrónico * campo requerido Por favor valida tu correo electrónico.

Teléfono * campo requerido

Teléfono trabajo

Celular * campo requerido

Figura 7 Prueba de envío de correo en inscripción exitosa

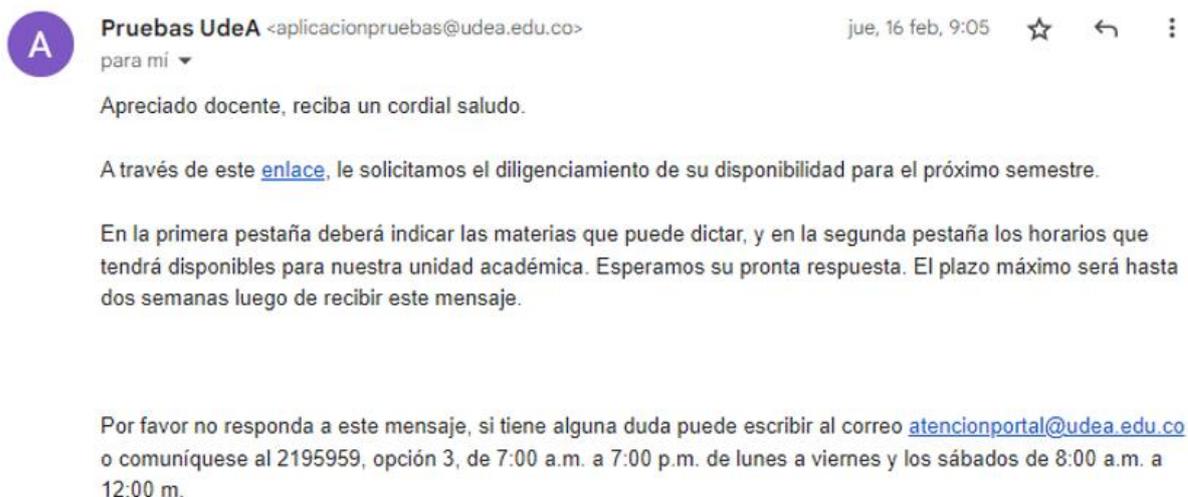


Figura 8 Lista de requisitos en módulo de admisiones

Posgrados - Requisitos

UdeA

Ingrese la descripción del nuevo requisito:

Guardar
Limpiar
Regresar

UdeA

1	Anexo Hoja de Vida: Calificaciones de Pregrado	Actualizar Deshabilitar
2	Anexo Hoja de Vida: Experiencia Laboral	Actualizar Deshabilitar
	Anexo Hoja de Vida: Experiencia Investigativa como estudiante o como profesional	Actualizar Deshabilitar

Figura 9 Edición de resolución en módulo de admisiones

UdeA

Información de la resolución:

Nro Resolución:	133	Fecha Resolución:	14/12/2004
Vigencia	14/12/2005	Emite Resolución:	RCCP

UdeA

Asociar requisitos a programa

Programa	Id	Descripción	Solicitado	Insc.	Admis.
50012 - ESPECIALIZACIÓN EN DERE...	99	Diploma o Acta de Grado de título profesional en Derecho	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Enviar
Regresar

Figura 10 Búsqueda de programa por código o nombre en módulo de admisiones

UdeA

Usuario del port...

Correo :

Codigo

Buscar programa

UdeA

Texto o código a buscar 50 **Terminar**

Codigo	Nombre	Añadir
70050	DOCTORADO EN CIENCIAS ...	+
50235	ESP EN CIENCIAS BÁSICAS ...	+
50088	ESP EN GERENCIA INTEGRA...	+
50224	ESP. CLÍNICA EN ODONTOL...	+
50272	ESP. EN CUIDADO DE ENFER...	+
50191	ESP. EN EDUCACIÓN FCA: A...	+
50192	ESP. EN EDUCACIÓN FCA: A...	+

+

Figura 11 Lista de programas con revisor asignado nombre en módulo de admisiones

UdeA

Lista de programas con usuario

Programa	Correo	Usuario	
DOCT. CCIAS BÁS BIOMÉD:ENF BIOQUÍMICA FARMACOLOG	andres.naranjo@udea.edu.co	andres.naranjo	🗑
DOCT. CIENCIAS BÁS. BIOMÉD:ÉNFASIS MICROBIOLOGÍA Y I	andres.naranjo@udea.edu.co	andres.naranjo	🗑
DOCT. CIENCIAS BÁSICAS BIOMÉDICAS:ÉNFASIS GENÉTICA	mmartinezg@udea.edu.co	marlen.martinez	🗑
DOCTORADO CIENCIAS FARMACÉUTICAS Y ALIMENTARIAS	posgradoqf@quimbaya.udea.edu.co	dora.benjumea	🗑
DOCTORADO CIENCIAS FARMACÉUTICAS Y ALIMENTARIAS	arley.patino@udea.edu.co	arley.patino	🗑
DOCTORADO CIENCIAS FARMACÉUTICAS Y ALIMENTARIAS	posgradoqf@quimbaya.udea.edu.co	dora.benjumea	🗑
DOCTORADO CIENCIAS FARMACÉUTICAS Y ALIMENTARIAS	arley.patino@udea.edu.co	arley.patino	🗑
DOCTORADO EN AGROECOLOGÍA	diana.florez@udea.edu.co	diana.florez	🗑
DOCTORADO EN ARTES	amparo.barrera@artes.udea.edu.co	amparo.barrera	🗑
DOCTORADO EN BIOTECNOLOGÍA	zmonzalve@gmail.com	zulma.monsalve	🗑

Items per page: 10 1 - 10 of 396 < > >>

8 Pruebas

El proceso de pruebas en la división de gestión informática se lleva a cabo de una manera específica, primero se sube los aplicativos a un servidor dedicado para desarrollo, esto con el fin de que el aplicativo este configurado adecuadamente para que se instale y corra en el servidor, luego se hace una petición de despliegue en el servidor de pruebas con los pasos a seguir para su instalación y funcionamiento verificados durante el paso anterior de instalación en el server desarrollo. Una vez los aplicativos están en el servidor de desarrollo y se verifica que estén corriendo adecuadamente se llevan a cabo pruebas de funcionalidad con ayuda del ingeniero encargado del desarrollo anterior, esto requiere sincronizarse con el dueño del esquema de la base de datos ya que se necesitan habilitar datos para los diferentes casos de prueba, datos no solo del aspirante sino también las resoluciones y los programas así como ajustar los tiempos de calendario de estos para poder realizar las pruebas de inscripción.

Durante las pruebas el objetivo era encontrar errores y bugs que se podrían haber pasado por alto durante el proceso de desarrollo y las pruebas locales de funcionamiento a pequeña escala que se realizaban a medida que avanzaba el

desarrollo, también podrían introducirse cambios en el proceso según se considerara conveniente.

Luego de las pruebas de funcionamiento también se realizó una prueba de concepto ante el área de admisiones y registro encargada de posgrado para darles a conocer el aplicativo, así como también conocer su opinión y escuchar sugerencias sobre el funcionamiento. Los resultados se analizaron y se realizaron las correcciones y cambios necesarios, aunque aún son necesarias más pruebas para su salida a producción.

Al final se agendó una prueba más ante un equipo de desarrollo encargado de un proyecto que estuvo archivado durante un tiempo, este proyecto se trata de una propuesta de unificación de las inscripciones de posgrado y pregrado, y el cual implica varios cambios entre los cuales el más importante es la eliminación del pago previo como requisito para la inscripción y que en cambio este se realice posteriormente al diligenciamiento del formulario de inscripción. Estos cambios introducen modificaciones al aplicativo encargado de la inscripción del aspirante que ya se están analizando con el equipo encargado de este proyecto, así como también con la arquitecta y el ingeniero encargados del anterior aplicativo de inscripción posgrados y el dueño del esquema de la base de datos

9 Documentación, Implementación, Mantenimiento Y Soportes

Para la documentación en la división de gestión informática se tiene como práctica documentar en el código (sobretudo el backend Java) las clases, entidades, métodos y sus variables para facilitar el mantenimiento a futuro de los aplicativos, también se desarrolló un manual técnico donde se documenta las tecnologías usadas, el proceso y la persistencia de los aplicativos. En el caso del web service wsRedoc se deja una ficha técnica para los analistas y otra para los clientes, en estas se detalla los datos de que recibe y los que entrega el servicio, así como también detalles del proceso en la ficha del analista.

En cuanto a la implementación cada uno de los módulos tiene una implementación diferente y se avanza de manera distinta.

El módulo de inscripción de aspirantes es un módulo único que sigue un flujo de datos específico.

Por otra parte, el módulo administrativo se compone de varios procesos aislados lo cual permite un desarrollo más modular, en este se fue aislando los procesos uno a uno y se fue realizando pruebas sobre los módulos terminados en paralelo al desarrollo de los siguientes, debido a que el proceso y la persistencia de datos se mantuvo igual que en el aplicativo anterior se tiene la posibilidad de desplegar los procesos conforme se vayan terminando sin afectar el gran esquema de la admisión de aspirantes de posgrado. De este módulo se desarrollaron completamente varios sub-módulos y se realizaron las pruebas sobre estos. Este módulo está en proceso de despliegue en producción realizando los cambios necesarios en el servidor para desplegar aplicaciones JPA. En cuanto a la prueba de concepto del uso de JPA se ha avanzado de manera satisfactoria y no se ha generado ningún entorpecimiento del proceso ni problemas con la persistencia de los datos por lo cual se espera que se pueda utilizar en más aplicaciones de la división en un futuro.

Por último y como se comentó en un principio, parte del desarrollo de la práctica también implicaba el soporte de algunas aplicaciones existentes que fueron asignadas. El soporte generalmente consistía en atender y corregir fallos en dichas aplicaciones los cuales se informaban por medio del sistema de tickets de la universidad, aunque también se presentaban casos en los cuales se debía implementar nuevas funcionalidades sobre estos las cuales se atendieron oportunamente.

10 Resultados Finales

Teniendo en cuenta los resultados obtenidos en las pruebas del módulo de inscripción de aspirantes se evidencia que este funciona de la manera esperada:

- Las validaciones y el flujo del proceso funcionan correctamente direccionando al aspirante acorde a el punto del proceso en que se encuentra y los datos que tiene.
- Los cambios aplicados hasta el momento se han incorporado sin afectar el proceso de inscripción.
- Al utilizar JDBC no se induce un cambio sustantivo a la persistencia, pero si se facilita el mapeo de las entidades y parece que el llamado de funciones y procedimientos almacenados desde la base de datos es más eficiente.
- La implementación del frontend en Angular facilita la separación de responsabilidades, analizando correctamente el código de la aplicación anterior se puede separar este en sus correspondientes componentes en la nueva implementación de forma tal que la aplicación llega a más sentido separando la lógica del negocio en el back y las acciones del usuario en el front de manera más congruente.
- Los llamados http desde el front hacia el back funcionan correctamente, sin embargo, se evidencia que se podría compactar de mejor forma el manejo de los datos que regresan los llamados en la lógica typeScript, es algo que con mejor entendimiento se mejoró en el módulo de admisiones y que se planea analizar e ir implementando en el des aspirante para compactar el código.
- Haciendo uso de las opciones de memoria de la aplicación que facilita angular y combinando adecuadamente sessionStorage, localStorage y el almacenamiento de datos temporales en los servicios, se puede mejorar la seguridad en el manejo de datos en cada sesión de los usuarios.
- La persistencia de datos funciona correctamente sin ningún problema o incongruencia.

En cuanto a el módulo de Admisiones:

- Se evidencia que separar las aplicaciones fue una decisión acertada ya que facilita la mantenibilidad de la aplicación. Se podría plantear la posibilidad de separar en diferentes módulos este aplicativo ya que al constar de procesos que son en cierto modo independientes entre sí, se podría utilizar como caso de prueba hacia una implementación totalmente RESTfull.
- La implementación de JPA facilita el proceso de desarrollo y no parece tener un impacto negativo en la persistencia.
- La implementación en el servidor de pruebas demuestra que el despliegue progresivo de sub-modulos según se vayan desarrollando es posible sin generar inconsistencia en los datos.
- Al momento de desplegar en el servidor de producción no hay inconsistencias más allá de un par de ajustes necesarios que ya se hablaron con el proveedor.

11 Dificultades

- Como en todo proceso de desarrollo realizado por un estudiante en medio de sus prácticas académicas se presentaron dificultades e imprevistos propios de la inexperiencia, estos se sortearon con la ayuda de los asesores y desarrolladores del equipo de desarrollo web pertenecientes a la división de gestión de informática de la Universidad de Antioquia.
- Tal vez la dificultad mayor y de la cual se desprenden muchas de las otras es inherente a cómo se manejan los procesos en la Universidad de Antioquia, pese al interés y disposición del equipo de desarrollo web y de la división de gestión informática las comunicaciones necesarias con otras áreas y la jerarquía de los procesos al ser esta una entidad pública tiende a ralentizar un poco las cosas.
- El desarrollo paralelo que se está realizando en otra división para el área de admisiones y registro que interviene parte del proceso del módulo administrativo y la comunicación con esta (ver dificultad anterior) llevó al replanteo en varias oportunidades del scope y la estructura del proceso, llegando incluso a el descarte de la implementación del sistema reDoc por esta misma causa.
- Al ser una migración de un desarrollo actualmente ya implementado esta conlleva una serie de retos y dificultades propios de este proceso, entender y analizar no solo la funcionalidad si no también el código anterior trae consigo una serie de retos y enseñanzas particulares.
- En cuanto al uso del API JPA para la persistencia se presentaron dificultades de compatibilidad a la hora de desplegar en los servidores.
- También por el uso de la API anteriormente mencionada se

12 Conclusiones

Con respecto al trabajo realizado durante este proceso de práctica académica se puede concluir que:

- Se debe tener mucha precaución con el manejo y las validaciones de los datos de los aspirantes a la hora de plantear y desarrollar la lógica del proceso de inscripción.
- Aunque ambos módulos dependen del mismo esquema y tablas en base de datos, se puede separar inscripción y revisión sin generar afectaciones en uno u en otro.
- El análisis previo de los procesos resultó fundamental a la hora de reconocer cuándo es posible y cuándo no separar en sub-módulos un aplicativo y desplegarlo de manera paulatina.
- Al momento de utilizar JPA hay que tener muy en cuenta las necesidades del proyecto, así como una constante comunicación con el dueño del esquema de base de datos y los proveedores del servidor.
- También para el uso de JPA es indispensable el análisis constante de los llamados a base de datos, así como cuando es prudente crear una entidad para el mapeo de datos y cuando un llamado JPQL o SQL nativo es una buena opción. En cuanto a esto último también es necesario plantearse si JPA es la solución adecuada para el proceso, sobre todo cuando se empieza a encontrar la necesidad frecuente de llamados nativos y mapeos forzados de entidades.
- Los conocimientos técnicos y de lógica, así como la ética profesional inculcada durante el proceso de formación en la Universidad resultan indispensables al momento de manejar un proceso de tal delicadeza como lo es la inscripción de aspirantes que resulta ser crítico para esta asegurar la consistencia en los datos y la seguridad de los mismos.
- Este proceso se logró llevar a cabo gracias a el acompañamiento de los asesores y el soporte de todo el equipo de la división de gestión informática.

13 Referencias Bibliográficas

Arquitectura REST: la arquitectura del momento. In gausswebapp.com, 2014.

Available: <https://gausswebapp.com/arquitectura-rest.html>.

Cano C. La Arquitectura REST. In: tsgroup.com.co, 2017. Available:

[http://www.tsgroup.com.co/wps/portal/tsg/blog/detalle-blog/la-arquitectura-rest#:~:text=El%20estilo%20REST%20hace%20%C3%A9nfasis,recursos%20universales%20%C3%BAnicos%20\(URI\).](http://www.tsgroup.com.co/wps/portal/tsg/blog/detalle-blog/la-arquitectura-rest#:~:text=El%20estilo%20REST%20hace%20%C3%A9nfasis,recursos%20universales%20%C3%BAnicos%20(URI).)

Java (lenguaje de programación). In: wikipedia.org. Available:

[https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

Java Persistence Query Language. In wikipedia.org. Available:

https://es.wikipedia.org/wiki/Java_Persistence_Query_Language

Java. In <https://desarrolloweb.com/>. Available:

<https://desarrolloweb.com/home/java>.

JavaServer Faces(JSF). In juntadeandalucia.es. Available:

<https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/101>

Muradas Y. Qué es Spring Framework y por qué usarlo. In openwebinars.net,

2018. Available: <https://openwebinars.net/blog/conoce-que-es-spring-framework-y-por-que-usarlo/>.

Pahino R. ¿Qué son Spring framework y Spring Boot? Tu primer programa

Java con este framework. In campusmvp.es, 2020. Available:

<https://www.campusmvp.es/recursos/post/que-son-spring-framework-y-spring-boot-tu-primer-programa-java-con-este-framework.aspx>.

Universidad de Antioquia. (2022). Contenido institucional, División de Gestión Informática. Available:

<http://www.udea.edu.co/wps/portal/udea/institucional/direccion-planeacion-desarrollo-institucional/division-gestion-informatica>

¿Qué es una API?. In: redhat.com. 2017. Available:

<https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces#:~:text=Una%20API%20o%20interfaz%20de,el%20software%20de%20las%20aplicaciones>.