

Aplicación web para gestionar reseñas estudiantiles sobre profesores de educación superior

José Saúl Vega Arrieta

Trabajo de grado para Optar el Título de Ingeniero de Sistemas e Informática

Director

Sonia Cristina Gamboa Sarmiento

Doctor en educación

Universidad Industrial Santander

Facultad de Ingenierías Físico-Mecánicas

Escuela de Ingeniería de Sistemas

Bucaramanga

2021

**Tabla de Contenido**

Introducción ..... 11

1. Planteamiento y justificación del problema ..... 12

2. Objetivos ..... 13

2.1. Objetivo general ..... 13

2.2. Objetivos específicos ..... 13

3. Marco de referencia ..... 14

3.1. Marco contextual ..... 14

3.1.1. Evaluación docente ..... 14

3.1.2. Importancia de las evaluaciones docente ..... 14

3.1.3. Mitos versus investigación sobre evaluación docente..... 15

3.1.4. Aplicación de una sola página (SPA) ..... 17

3.1.5. API REST..... 18

3.1.6. Autenticación con token..... 18

3.1.7. Websockets..... 19

3.1.8. Framework ..... 19

3.1.9. Feed de usuario..... 19

3.2. Marco tecnológico ..... 20

3.2.1. Visual studio code..... 20

3.2.2. Javascript..... 20

3.2.3. EcmaScript 6 (ES6) ..... 20

3.2.4. ReactJS..... 21

3.2.5. NodeJS ..... 22

3.2.6. Create-react-app..... 22

3.2.7. React-router ..... 22

3.2.8. Bulma ..... 22

3.2.9. Redux ..... 23

3.2.10. React-redux ..... 23

3.2.11. Python ..... 23

3.2.12. Django..... 24

3.2.13. Django rest framework..... 24

3.2.14. Djoser ..... 25

3.2.15. Django channels..... 25

3.2.16. Redis ..... 25

3.2.17. Postgresql ..... 25

3.2.18. Amazon Web Services ..... 26

3.2.19. Elastic beanstalk ..... 26

3.2.20. Elastic Compute Cloud (EC2) ..... 27

3.2.21. Relational Database Service (RDS) ..... 27

3.2.22. Simple Storage Service (S3)..... 28

3.3. Marco de arquitectura software ..... 28

3.3.1. Modelo-vista-controlador (MVC)..... 28

4. Estado del arte ..... 29

4.1. Ratemyprofessors.com..... 29

4.2. Ratemyteachers.com ..... 30

4.3. DetodoUIS ..... 31

5. Metodología ..... 32

6. Desarrollo del proyecto..... 33

6.1. Análisis y requerimientos..... 33

6.1.1. Requerimientos funcionales ..... 34

6.1.2. Requerimientos no funcionales ..... 44

6.2. Diseño del software ..... 46

6.2.1. Casos de uso ..... 46

6.2.2. Diseño de la interfaz gráfica..... 51

6.2.3. Diagramas de actividades ..... 68

6.2.4. Modelo de base de datos ..... 71

6.3. Arquitectura del software ..... 71

6.3.1. Modelo-vista-controlador (MVC)..... 72

6.3.2. Modelo ..... 72

6.3.3. Vista ..... 72

6.3.4. Controlador..... 72

6.4. Implementación de los prototipos..... 73

6.4.1. Primer prototipo..... 73

6.4.2. Segundo prototipo..... 79

6.5. Pruebas..... 81

6.5.1. Pruebas del primer prototipo ..... 81

6.5.2. Pruebas del segundo prototipo..... 93

6.6. Configuraciones en Amazon web services..... 94

6.6.1. Aplicación de React en S3 ..... 95

6.6.2. Aplicación de Django en Elastic Beanstalk.....	97
7. Conclusiones .....	98
8. Recomendaciones .....	99
Referencias Bibliográficas .....	100

**Lista de tablas**

Tabla 1. Requerimiento funcional: registrar usuario ..... 34

Tabla 2. Requerimiento funcional: iniciar sesión..... 34

Tabla 3. Requerimiento funcional: cerrar sesión..... 35

Tabla 4. Requerimiento funcional: edición de cuenta de usuario..... 35

Tabla 5. Requerimiento funcional: recuperación de contraseña ..... 36

Tabla 6. Requerimiento funcional: gestión de universidades ..... 36

Tabla 7. Requerimiento funcional: gestión de escuelas..... 37

Tabla 8. Requerimiento funcional: gestión de profesores..... 37

Tabla 9. Requerimiento funcional: gestión de usuarios ..... 37

Tabla 10. Requerimiento funcional: gestión de reseñas ..... 38

Tabla 11. Requerimiento funcional: guardar reseñas favoritas ..... 38

Tabla 12. Requerimiento funcional: compartir reseñas ..... 39

Tabla 13. Requerimiento funcional: recomendar reseña ..... 39

Tabla 14. Requerimiento funcional: consultar reseñas recomendadas ..... 40

Tabla 15. Requerimiento funcional: suscripción a entidades..... 40

Tabla 16. Requerimiento funcional: ver notificaciones ..... 41

Tabla 17. Requerimiento funcional: búsqueda de entidades..... 41

Tabla 18. Requerimiento funcional: visualizar reseñas de entidades ..... 42

Tabla 19. Requerimiento funcional: visualizar reseñas más recomendadas ..... 42

Tabla 20. Requerimiento funcional: visualizar últimas escuelas agregadas ..... 43

Tabla 21. Requerimiento funcional: ver profesores probables..... 43

Tabla 22. Requerimiento funcional: reportar reseña ..... 43

Tabla 23. Requerimiento funcional: feed de usuario..... 44

Tabla 24. Requerimiento no funcional: interfaz gráfica responsiva..... 45

Tabla 25. Requerimiento no funcional: interfaz gráfica SPA ..... 45

Tabla 26. Requerimiento no funcional: compatibilidad con navegadores ..... 45

Tabla 27. Descripción de actor visitante..... 48

Tabla 28. Descripción de actor usuario registrado ..... 49

Tabla 29. Descripción de actor administrador ..... 50

Tabla 30. Prueba iniciar sesión ..... 81

Tabla 31. Prueba registrar usuario..... 82

Tabla 32. Prueba cerrar sesión ..... 82

Tabla 33. Prueba cambio de nombre de usuario ..... 83

Tabla 34. Prueba cambio de correo ..... 83

Tabla 35. Prueba recuperación de contraseña ..... 84

Tabla 36. Prueba agregar universidades ..... 85

Tabla 37. Prueba agregar escuelas ..... 85

Tabla 38. Prueba agregar profesor ..... 86

Tabla 39. Prueba eliminar usuario..... 87

Tabla 40. Prueba agregar reseña ..... 87

Tabla 41. Prueba guardar reseña ..... 88

Tabla 42. Prueba compartir reseña ..... 88

Tabla 43. Prueba recomendar reseña..... 89

Tabla 44. Prueba consultar reseñas recomendadas..... 89

Tabla 45. Prueba suscribirse a entidades .....	90
Tabla 46. Prueba ver notificaciones .....	90
Tabla 47. Prueba búsqueda de entidades .....	90
Tabla 48. Prueba visualizar reseñas de entidades .....	91
Tabla 49. Prueba visualizar reseñas más recomendadas en inicio .....	91
Tabla 50. Prueba visualizar últimas escuelas agregadas .....	92
Tabla 51. Prueba ver profesores probables .....	92
Tabla 52. Prueba reportar reseña .....	93
Tabla 53. Prueba visualizar reportes.....	93
Tabla 54. Prueba feed de usuario .....	94

**Lista de Figuras**

Figura 1. Modelo de prototipos ..... 33

Figura 2. Casos de uso ..... 47

Figura 3. Diseño del logo..... 51

Figura 4. Diseño de cabecera ..... 52

Figura 5. Diseño de menú principal ..... 52

Figura 6. Diseño componente profesor en menú principal ..... 53

Figura 7. Diseño de componente profesor en lista ..... 53

Figura 8. Diseño de componente detalles de profesor ..... 54

Figura 9. Diseño de componente detalles de universidad ..... 55

Figura 10. Diseño de componente escuela en lista ..... 55

Figura 11. Diseño de componente detalles de escuela..... 56

Figura 12. Diseño de componente universidad en lista ..... 56

Figura 13. Diseño de componente detalles de universidad ..... 57

Figura 14. Diseño de componente reseña ..... 57

Figura 15. Diseño de componente reseña en resultados de búsqueda ..... 58

Figura 16. Diseño de componente reseña en menú principal..... 58

Figura 18. Diseño de página principal ..... 60

Figura 19. Diseño de lista de instituciones ..... 60

Figura 20. Diseño de lista de profesores ..... 61

Figura 21. Diseño de resultados de búsqueda de profesores..... 61

Figura 22. Diseño de resultados de búsqueda de universidades ..... 62

Figura 23. Diseño de perfil de instituciones ..... 62

Figura 24. Diseño de perfil de profesores ..... 63

Figura 25. Diseño de formulario creación de reseñas ..... 63

Figura 26. Diseño de biblioteca de usuario..... 64

Figura 27. Diseño de feed usuario ..... 64

Figura 28. Diseño de reseñas de usuario..... 65

Figura 29. Diseño de notificación de usuario..... 65

Figura 30. Diseño de edición de datos de usuario ..... 66

Figura 31. Diseño de compartir reseña ..... 66

Figura 32. Diseño de formulario de registro ..... 67

Figura 33. Diseño de inicio de sesión ..... 67

Figura 34. Diagrama de actividades para administración de entidades ..... 68

Figura 35. Diagrama de actividades para gestión de reseñas ..... 69

Figura 36. Diagrama de actividades para búsqueda de entidades ..... 70

Figura 37. Modelo de base de datos ..... 71

Figura 38. Arquitectura del software ..... 73

Figura 39. Vista de escritorio y móvil de registro de usuario ..... 73

Figura 40. Vista de escritorio y móvil de inicio de sesión ..... 74

Figura 41. Vista de escritorio y móvil de página principal ..... 74

Figura 42. Vista de escritorio y móvil de resultados de búsqueda ..... 75

Figura 43. Vista de escritorio y móvil de perfiles de entidades ..... 75

Figura 44. Vista de escritorio y móvil de creación de reseña ..... 76

Figura 45. Vista de escritorio y móvil de reseñas de usuario..... 76

Figura 46. Vista de escritorio y móvil de edición de reseñas.....	77
Figura 47. Vista de escritorio y móvil de notificaciones de nueva reseña .....	77
Figura 48. Vista de escritorio y móvil de edición de cuenta .....	78
Figura 49. Vista de escritorio y móvil de biblioteca.....	78
Figura 50. Vista de escritorio y móvil de feed de usuario .....	79
Figura 51. Vista de escritorio y móvil de página 404.....	79
Figura 52. Vista de escritorio y móvil para reportar reseña .....	80
Figura 53. Vista de escritorio y móvil de esqueletos de carga.....	80

## Resumen

**Título:** Aplicación web para gestionar reseñas estudiantiles sobre profesores de educación superior\*

**Autor:** José Saúl Vega Arrieta\*\*

**Palabras claves:** reseñas de estudiantes, reseñas sobre profesores, creación de reseñas.

**Descripción:** Teniendo en cuenta la necesidad de estudiantes universitarios por conocer de antemano los profesores con quienes darán clases, se propone crear una aplicación web que permita la creación de reseñas sobre profesores por parte de estudiantes universitarios, la cual permita a otros estudiantes conocer sobre sus futuros profesores. Aunque otras herramientas tecnológicas se han usado con el mismo fin, estas no ofrecen una forma eficaz de encontrar las opiniones y profesores que se buscan en el menor tiempo posible, al no estar especializadas en esta clase de requerimiento. Las diferentes herramientas software de hoy en día permiten una creación rápida y organizada de aplicaciones de este tipo sin el enorme requerimiento de grandes recursos y gran personal; herramientas como ReactJS y Django se pueden aprender rápidamente y comenzar a usar fácilmente con muchos beneficios, ya que estas tienen dentro de su estructura funcionalidades que siempre son requeridas para la creación de aplicaciones web. La arquitectura MVC usada por esta clase de herramientas o frameworks llega a ser útil para separar vistas gráficas (frontend) de la lógica del servidor (backend), y es que con el tiempo estos módulos independientes pueden crecer en tamaño dificultado el orden entre equipos desarrolladores de la aplicación web.

\* Aplicación web para gestionar reseñas estudiantiles sobre profesores de educación superior.

\*\* Facultad de Ingenierías Físico-Mecánica. Escuela de Ingeniería de Sistemas e Informática.  
Director: Sonia Cristina Gamboa Sarmiento.

### Abstract

**Title:** Web application to manage student reviews of higher education teachers\*

**Author:** José Saúl Vega Arrieta\*\*

**Keywords:** student reviews, teacher reviews, review creation.

**Description:** Taking into account the need of university students to know in advance the professors with whom they will teach, it is proposed to create a web application that allows the creation of reviews about professors by university students, which allows other students to know about their future professors . Although other technological tools have been used for the same purpose, they do not offer an efficient way to find the opinions and teachers that are sought in the shortest possible time, as they are not specialized in this kind of requirement. The different software tools of today allow a fast and organized creation of applications of this type without the enormous requirement of great resources and great personnel; Tools like ReactJS and Django can be learned quickly and easily started with many benefits, since they have within their structure functionalities that are always required for the creation of web applications. The MVC architecture used by this class of tools or frameworks becomes useful to separate graphical views (frontend) from the server logic (backend), and it is that over time these independent modules can grow in size, making order between development teams of the web application difficult.

\* Aplicación web para gestionar reseñas estudiantiles sobre profesores de educación superior.

\*\* Facultad de Ingenierías Físico-Mecánica. Escuela de Ingeniería de Sistemas e Informática.

Director: Sonia Cristina Gamboa Sarmiento.

## Introducción

Los estudiantes quieren saber quién les dictará clases, conocer un poco del profesor que les han asignado, su metodología y dificultad a lo largo del semestre; información clave para todo estudiante universitario. Hoy en día las tecnologías que facilitan la comunicación, más que todo redes sociales, son usadas para intercambiar esta información, pero la búsqueda de opiniones sobre profesores es difícil ya que estos sitios no están especializados en esto, y están llenos de información no relevante. Con esto en mente, se propone la creación de una aplicación web de reseñas que permita a estudiantes universitarios escribir su opinión sobre los profesores que han visto y permita a otros estudiantes leerlas.

La universidad es un sitio donde se reúnen varios elementos, los más importantes siendo los estudiantes y profesores, y las escuelas o departamentos donde interactúan. La universidad es el elemento más grande que todo lo engloba, y las escuelas son el mínimo elemento dentro de cada universidad. Los profesores pertenecen a una universidad y a su vez están inscritos en alguna escuela. Es esta jerarquía la que se va a plasmar en los datos del sitio de reseñas de manera que se pueda ajustar fácilmente a la forma en la que están organizadas la mayoría de universidades.

Aplicando la metodología de modelo de prototipos, la cual implica el desarrollo rápido del prototipo con uso de pocos recursos, se usará el framework de interfaces de usuario ReactJS de Javascript que cuenta con poco peso y es de fácil aprendizaje y manejo. Junto con el framework Django del lenguaje de programación Python, especializado en la creación rápida de aplicaciones web.

Al final, con esta clase de aplicación web se espera que se le facilite al estudiante universitario hacerse una idea del profesor que le dará clases en el futuro inmediato, con ayuda de compañeros estudiantes que tienen experiencia con el profesor. Contando con un buen

sistema de búsqueda también se espera que los usuarios de la aplicación puedan, solo con el nombre, encontrar fácilmente la universidad o los profesores sobre los cuales quieran opinar o sobre los cuales quieran saber.

### **1. Planteamiento y justificación del problema**

Al inicio de cada semestre académico, los estudiantes universitarios se ven en la necesidad de indagar sobre los profesores cuyos cursos están en la posibilidad de matricularse. Principalmente, se preguntan sobre la metodología de enseñanza, la dinámica de las clases, los métodos de evaluación y algún nivel de dificultad, elementos sobre los cuales pueden dar razón estudiantes que ya han cursado tales materias. Habitualmente, los estudiantes realizan búsquedas en redes sociales o preguntan entre sus contactos por compañeros estudiantes que ya tengan experiencia con los profesores, pero esta información no queda registrada para ser luego consultada con este propósito, no puede ser consultada con filtros de búsqueda y, por consiguiente, generalmente se pierde la memoria recopilada sobre la experiencia de los estudiantes con los profesores. Teniendo en cuenta lo anterior la creación de una aplicación web que permita registrar esta información, de manera estructurada, por parte de estudiantes universitarios, para ser consultada también por estudiantes universitarios, es necesaria.

## 2. Objetivos

### 2.1. Objetivo general

Diseñar y desarrollar una aplicación web para recopilar reseñas de estudiantes sobre profesores de instituciones de educación superior.

### 2.2. Objetivos específicos

- Dar a conocer la relevancia de las calificaciones de estudiantes sobre profesores.
- Analizar los beneficios de las evaluaciones docente en el proceso educativo.
- Determinar los requerimientos funcionales y no funcionales para la aplicación.
- Diseñar los componentes software para la aplicación.
- Desarrollar una aplicación web de reseñas con las siguientes funcionalidades:
  - Registro de usuario.
  - Creación de reseña.
  - Calificación de reseña.
  - Enviar notificación de nueva reseña al usuario.
  - Agregar universidad.
  - Agregar escuela/facultad.
  - Agregar profesor.
  - Búsqueda de universidad, escuela y profesor.
- Publicar la aplicación web para uso de la comunidad estudiantil.

### 3. Marco de referencia

#### 3.1. Marco contextual

Para el mejor entendimiento del siguiente proyecto se exponen y explican los siguientes conceptos relevantes.

**3.1.1. Evaluación docente.** La evaluación docente o calificaciones de estudiantes, son evaluaciones hechas por estudiantes sobre un curso o profesor específico; por lo general se trata de una encuesta que se completa al final del curso. La encuesta está diseñada teniendo en cuenta la intención de las instituciones para la evaluación (igi-global.com, s.f).

**3.1.2. Importancia de las evaluaciones docente.** Existen muchas opiniones sobre las evaluaciones docente, desde completo rechazo a aceptación y uso de estas. Aun si, la realimentación estudiantil es importante. Se trata de las opiniones de los estudiantes sobre su salón de clase. La gran mayoría de instituciones de educación superior usan evaluaciones docentes. Existe una variedad de usos para estas evaluaciones, como tomar decisiones para promociones o para medir la efectividad del profesor. Además, las evaluaciones ayudan al profesor a mirar su curso desde una perspectiva diferente, la perspectiva del estudiante. Las evaluaciones pueden ser de mucha ayuda si además de solo entregarlas, el profesor lee y considera los comentarios de estas. Los profesores deberían racionalmente considerar todos los comentarios tanto positivos como negativos y realizar los cambios necesarios. Pero, algunos profesores se pueden tomar las evaluaciones muy personales y ponerse a la defensiva. Esto puede crear desconfianza en las evaluaciones docente.

Los profesores que tienen una actitud positiva con el incremento del uso de las evaluaciones docente indican que las evaluaciones satisfacen el derecho de los estudiantes a ser escuchados, y estos profesores se esfuerzan por tener en cuenta los comentarios. Los profesores han tenido reacciones positivas sobre las realimentaciones que han recibido, indicando que ayudan al proveerlos de una justificación para cambios en el estilo de enseñanza y contenido del curso. Evaluaciones formativas que piden al estudiante que hagan cualquier pregunta que tengan en cuanto al curso, califiquen el desempeño del instructor y comenten sobre formas en la que se puede mejorar el curso; ayudan a los estudiantes a darse cuenta de la importancia de sus aportes, lo que resulta en una mayor participación en la clase y una mejor comunicación entre estudiantes y profesores. Además, los profesores indican que comentarios positivos en evaluaciones ayudan a sentirse más confiados, no solo en sus habilidades como profesor, sino también con su identidad personal (Mariano et al., 2018).

**3.1.3. Mitos versus investigación sobre evaluación docente.** Las evaluaciones docente por parte de estudiantes son objeto de debate en campus universitarios. Desafortunadamente, estos debates a menudo no están informados por la extensa investigación sobre este tema. La investigación sobre las evaluaciones docente muestra que los datos de estas son: multidimensionales; confiables y estables; principalmente una función del profesor que imparte el curso; relativamente poco afectado por una variedad de variables hipotéticas como sesgos potenciales; y útiles para los profesores, estudiantes y administradores. Las siguientes son preguntas que suelen plantearse sobre el tema.

a) **¿Están los estudiantes calificados para evaluar a sus profesores?**

Generalmente la respuesta es sí. Los estudiantes pueden informar sobre el comportamiento de los profesores, la cantidad de trabajo requerido, cuánto sienten que han aprendido y la dificultad del material. Pueden responder preguntas sobre la calidad de las lecturas del curso, la importancia de las tareas, la claridad del profesor, la disponibilidad y servicio del profesor, y otros aspectos que hacen parte del proceso de aprendizaje. Nadie está más calificado para informar lo que sucedió durante el semestre, simplemente porque nadie más está allí durante el semestre. Ciertamente los estudiantes están calificados para expresar su satisfacción o insatisfacción con su experiencia. Tienen derecho a expresar sus opiniones en cualquier caso, y nadie más puede informar hasta qué punto la experiencia fue útil, productiva, informativa, satisfactoria o valiosa. Aunque las opiniones sobre estos aspectos del curso no son una medida directa del desempeño del profesor o el curso aprendido, son indicadores legítimos de la satisfacción del estudiante.

**b) ¿Las evaluaciones se basan solamente en popularidad?**

No hay base para este argumento y no hay investigación para sustentarlo. Cuando surge este tema, el término popular nunca se define. La suposición de que la popularidad significa falta de conocimiento o dificultad es infundada. De hecho, varios estudios muestran que los estudiantes aprenden más en cursos en los que los maestros demuestran interés o preocupación por los estudiantes y su aprendizaje; por supuesto, estos profesores también reciben calificaciones más altas.

**c) ¿La evaluación docente se relaciona con el aprendizaje?**

El criterio más aceptable para una buena enseñanza es el aprendizaje de los estudiantes. Existen correlaciones altas entre la evaluación de los estudiantes de lo aprendido en un curso y las evaluaciones generales de los estudiantes del profesor del curso. Aún más reveladores son los estudios en cursos de varias secciones que emplearon un examen final común. En general, las evaluaciones de los estudiantes fueron las más altas para los profesores cuyos estudiantes obtuvieron mejores resultados en los exámenes. Estos estudios son la evidencia más sólida de la validez de las calificaciones de los estudiantes porque conectan las calificaciones con el aprendizaje.

**d) ¿Los estudiantes evalúan a los profesores de acuerdo a la nota obtenida?**

Esta es actualmente la pregunta más polémica en la investigación de evaluaciones docente. Existe evidencia consistente de una relación entre evaluaciones docente y notas: una correlación de alrededor de .20. Los estudios de validez de múltiples secciones proporcionan la evidencia más sólida de que las calificaciones reflejan el aprendizaje. Estos hallazgos llevan a la conclusión a la que llegan la mayoría de los investigadores: que debería haber una relación entre las evaluaciones docente y las notas porque la enseñanza eficaz conduce a un aprendizaje que conduce al logro y a la satisfacción de los estudiantes. Las evaluaciones docente simplemente reflejan esta secuencia (Theall, 2003).

**3.1.4. Aplicación de una sola página (SPA).** En este tipo de aplicaciones web, todo lo que se muestra y se procesa está dentro de la misma página, así que al pasar de una opción a otra no se recarga el navegador y es desde un único archivo HTML

(index.html) desde el que se reproduce absolutamente todo. Esto se logra al cargar todo el código HTML, JavaScript y CSS una sola vez y cargando dinámicamente los recursos necesarios como respuesta a las acciones del usuario. A veces esto tiene como consecuencia que el cargado inicial de las aplicaciones web tipo SPA pueda tardar en darse si contienen código bastante extenso.

En cuanto a la experiencia de usuario que ofrecen las SPA, al no recargar la página frente a cada acción del usuario la experiencia se vuelve mucho más agradable en comparación a las aplicaciones web de muchas páginas. Al no ser un código complejo, los datos pesan poco, por lo que las transferencias son increíblemente rápidas y la comunicación entre cliente y servidor es muy fluida (García, 2017).

**3.1.5. API REST.** La API de una aplicación web sirve de mediador entre el servidor y el cliente, y describe la manera en que los desarrolladores pueden acceder y recuperar información de una base de datos.

Una API de REST es un conjunto de reglas que permiten la comunicación entre aplicaciones web. REST utiliza el protocolo HTTP y muchas de sus características como parte de la estructura de su API. Cuando una API usa la arquitectura REST, se refiere simplemente al hecho de que las peticiones del cliente serán peticiones HTTP (POST, GET, PUT y DELETE) y las respuestas del servidor se darán en formato JSON o XML (Simoës, 2019).

**3.1.6. Autenticación con token.** Una forma de autenticarse en aplicaciones web. El usuario se autentica en nuestra aplicación, generalmente con un par usuario/contraseña. A partir de ahí, cada petición HTTP que haga el usuario va acompañada de un Token. Este Token no es más que una firma cifrada que permite a nuestro API identificar al

usuario. Este Token se almacena en el lado del cliente (ya sea por localStorage o sessionStorage) y el API es el que se encarga de descifrar ese Token y redirigir el flujo de la aplicación en un sentido u otro.

Como los tokens son almacenados en el lado del cliente, no hay información de estado y la aplicación se vuelve totalmente escalable. De esta forma se puede usar la misma API para diferentes aplicaciones (Web, Android, iOS) y solo debemos preocuparnos de enviar los datos en formato JSON y generar y descifrar tokens en la autenticación y posteriores peticiones HTTP a través de un middleware (Azaustre, 2015).

**3.1.7. Websockets.** Un WebSocket es un protocolo de comunicaciones que proporciona canales de comunicación “full-duplex” sobre una misma conexión TCP. Facilitan la transferencia de datos en tiempo real del servidor, y para el servidor. Siendo una comunicación bidireccional, el servidor puede enviar la información directamente al cliente en el momento de la conexión. Este envío de información en tiempo real es ideal para la implementación de notificaciones al usuario.

**3.1.8. Framework.** En informática, un framework es un software reutilizable para desarrollar rápidamente aplicaciones, ya sean para la web, de escritorio o móvil. En otras palabras, es una gran biblioteca con diferentes porciones de código reusable, una pieza de software que proporciona a los desarrolladores una base de código y formas consistentes y estandarizadas para crear sus aplicaciones sin necesidad de escribir código que ya ha sido implementado con anterioridad por otros (Ortiz, 2018).

**3.1.9. Feed de usuario.** Un feed es un flujo de contenido por el que se puede desplazar.

El contenido aparece en bloques parecidos que se repiten uno después del otro. Por ejemplo, un feed puede ser editorial (como una lista de artículos o noticias) o puede ser una ficha (una lista de productos, servicios, etc.). Los feeds pueden aparecer en cualquier lugar de la página (Google, s.f.).

### 3.2. Marco tecnológico

A continuación se describen las herramientas tecnológicas usadas durante el desarrollo del proyecto.

**3.2.1. *Visual studio code.*** Editor de código fuente ligero para escritorio disponible para los sistemas operativos Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución como .NET y Unity (Microsoft, s.f.).

**3.2.2. *Javascript.*** Lenguaje de programación de la web. JavaScript es un lenguaje de programación ligero e interpretado. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat. JavaScript es un lenguaje de programación basado en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos (Mozilla, 2020).

**3.2.3. *Ecmascript 6 (ES6).*** ECMAScript específicamente es el estándar que a partir del año 2015 a la actualidad se encarga de regir como debe ser interpretado y funcionar

el lenguaje JavaScript (proporciona las reglas, detalles y directrices que un lenguaje de scripting debe seguir). Su sexta versión fue una de las más grandes y significó muchas mejoras de las cuales se pueden mencionar la mejora de la sintaxis y actualización de la misma que trajo consigo símbolos y tipos de datos que no existían en las versiones anteriores, así como también mejoras a las estructuras de iteración. Con esta versión de ECMAScript librerías como ReactJS se desarrollan y trabajan (Muradas, 2020).

**3.2.4. *ReactJS*.** Se trata de una librería de JavaScript desarrollada por Facebook para construir interfaces de usuario. Ayuda a crear interfaces de usuario interactivas de forma rápida y sencilla. Se diseñan vistas que representen los datos que se deben mostrar para cualquier condición, y React se encargará de actualizar y renderizar los componentes correctos cuando las condiciones o datos cambien. Uno de los objetivos de la librería es crear una arquitectura de desarrollo más avanzada y ordenada. Una interfaz de usuario es básicamente creada a partir de un componente, el cual encapsula el funcionamiento y la presentación (Facebook Inc, s.f.).

En comparación con otras librerías de JavaScript, React tiene ciertas ventajas a tener en cuenta. Al tener las vistas asociadas a los datos, no se necesita escribir código para manipular la página cuando los datos cambian. Esta parte en otras librerías como jQuery es muy laboriosa de conseguir y es algo que React hace automáticamente.

Comparando con frameworks como es el caso de Angular o Ember, React no tiene el mismo alcance, pues no incluye todo lo que suele ofrecer un framework completo. Aun así, a partir de todo el ecosistema de la librería se puede llegar a las mismas características que tienen la mayoría de aplicaciones web, convirtiéndolo en una alternativa perfecta. Además, tiene mayor compatibilidad con otras librerías de JavaScript en comparación con otros frameworks del lenguaje (Alvarez, 2019).

**3.2.5. NodeJS.** Entorno de tiempo de ejecución de JavaScript. Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript. Transformó a JavaScript de algo que solo podía ejecutarse en el navegador en algo que se podría ejecutar en los ordenadores como si de aplicaciones independientes se tratara. Gracias a Node.js se puede ir un paso más allá en la programación con JavaScript no solo creando sitios web interactivos, sino teniendo la capacidad de hacer cosas que otros lenguajes de secuencia de comandos como Python pueden crear (Lucas, 2019).

**3.2.6. Create-react-app.** Es un ambiente cómodo para aprender React, y es la mejor manera de comenzar a construir una nueva aplicación de página única usando ReactJS. Create React App configura tu ambiente de desarrollo de forma que puedas usar las últimas características de Javascript, brindando una buena experiencia de desarrollo, y optimizando tu aplicación para producción. Para su instalación y uso se requiere tener también instalado NodeJS (Facebook Inc, s.f.).

**3.2.7. React-router.** La librería estándar para enrutamiento para ReactJS. React Router mantiene la interfaz de usuario sincronizada con la URL. Tiene una API simple con funciones poderosas como carga de código a medida que se necesite, coincidencia dinámica de rutas, útil para determinar las direcciones permitidas en un nuestro sitio e implementar redirecciones, y manejo de transición de ubicación integrado (Dabit, 2016).

**3.2.8. Bulma.** Librería CSS gratis y de código abierto usada por más de 200,000 desarrolladores. Tiene la filosofía de desarrollo mobile-first, por lo tanto se tiene en cuenta primero las pantallas más pequeñas (celulares y tablets) a la hora de crear el

estilo y posicionamiento de los componentes de la aplicación web, para después pasar a adaptar el estilo de los componentes a pantallas más grandes (computadoras de escritorio y portátiles) (Bulma, s.f.).

**3.2.9. *Redux*.** Patrón de arquitectura de datos que permite manejar el estado de la aplicación de una manera predecible. Está pensado para reducir el número de relaciones entre componentes de la aplicación y mantener un flujo de datos sencillo. Iniciado originalmente por la comunidad de ReactJS, Redux se ha convertido en un patrón transversal, capaz de adaptarse a cualquier tipo de librería o framework para JavaScript. Incluso se podría usar sin necesidad de otro framework Javascript. También se puede ejecutar del lado del servidor o en aplicaciones para móviles, por lo que sus ámbitos de aplicación son muy amplios (Alvarez, 2018).

**3.2.10. *React-redux*.** Mantenido por el mismo equipo de Redux, react-redux es una dependencia de ReactJS que facilita el enlace entre los componentes de la librería y Redux. Permite que los componentes React lean datos almacenados en Redux y envíen acciones para actualizar dichos datos (Abramov, 2015).

**3.2.11. *Python*.** Lenguaje de programación interpretado, interactivo y orientado a objetos. Incorpora módulos, excepciones, tipificación dinámica, tipos de datos dinámicos de muy alto nivel y clases. Admite múltiples paradigmas de programación más allá de la programación orientada a objetos, como la programación funcional y de procedimientos. Python combina una potencia notable con una sintaxis muy clara. También se puede utilizar como lenguaje de extensión para aplicaciones que necesitan una interfaz programable. Se puede ejecutar en muchas variantes de Unix, incluidos Linux y macOS, y en Windows (Python Software Foundation, 2001). En los últimos

años se ha convertido en uno de los lenguajes de programación más usados por la comunidad de desarrolladores a nivel mundial (Statista, 2020).

**3.2.12. *Django.*** Django es un framework web de Python que posibilita el desarrollo rápido de aplicaciones web usando un diseño pragmático. Escrito por desarrolladores experimentados se ocupa de muchos aspectos del desarrollo web, de manera que el programador se pueda enfocar en escribir su aplicación sin necesidad de “reinventar la rueda” para muchos problemas que ya fueron solucionados. El framework es gratis y de código abierto (Django Software Foundation, 2005).

Django fue diseñado para ayudar a desarrolladores a transformar aplicaciones de conceptos y requerimientos a un proyecto completado, lo más rápido posible. Haciéndose cargo de muchos aspectos como: la seguridad, previene a desarrolladores de cometer errores de seguridad muy comunes; enrutamiento, basta con declarar que direcciones URL son aceptadas y definir qué página debe mostrarse para tal dirección para direccionar al usuario al lugar deseado; construcción de la base de datos, facilita la creación de bases de datos mediante simple código Python usando un mapeo objeto-relacional (conocido como ORM por sus siglas en inglés), esto evita la escritura manual de código SQL (Structured Query Language) necesario para realizar las consultas a bases de datos.

**3.2.13. *Django rest framework.*** Dentro del ecosistema del framework Django existe una herramienta que facilita la creación de APIs que usan una arquitectura REST, Django REST framework. Este poderoso kit de herramientas agiliza la creación de autenticaciones de usuario entre cliente y servidor mediante tokens, la transformación de datos del servidor a simple formato JSON mediante serializadores, y otras

características que completan cualquier API REST.

**3.2.14. *Djoser.*** La biblioteca Djoser proporciona un conjunto de funciones para manejar acciones básicas como registro, inicio de sesión, cierre de sesión, restablecimiento de contraseña y activación de cuenta dentro de proyectos creados con Django que usen Django REST Framework para la creación de una API. El paquete funciona con un modelo de usuario personalizado y utiliza autenticación basada en Token ([django-rest-framework.org](https://django-rest-framework.org), s.f.).

**3.2.15. *Django channels.*** Channels de Django, es un proyecto, que queriendo ir más allá del HTTP, incluye el manejo de WebSockets y protocolos de chat para posibilitar la comunicación en tiempo real entre un cliente y servidor (Django Software Foundation, 2018).

**3.2.16. *Redis.*** Redis (REmote DIctionary Server), es un almacén de estructura de datos en memoria que se puede utilizar como base de datos, caché o intermediario de mensajes. Los datos se almacenan en Redis en forma de pares clave-valor donde las claves se utilizan para ubicar y extraer los datos almacenados en la instancia de Redis. Las bases de datos normales almacenan datos en el disco, lo que conlleva un costo adicional, en términos de tiempo y recursos de hardware. Redis evita esto al almacenar todos los datos en la memoria, lo que hace que los datos estén fácilmente disponibles y aumenta la velocidad de acceso y manipulación de los datos, en comparación con las bases de datos normales (Gori, 2020).

**3.2.17. *Postgresql.*** Potente sistema de base de datos relacional de código abierto que

utiliza y amplía el lenguaje SQL combinado con muchas funciones que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. Los orígenes de PostgreSQL se remontan a 1986 como parte del proyecto POSTGRES en la Universidad de California en Berkeley y tiene más de 30 años de desarrollo activo en la plataforma central.

PostgreSQL se ha ganado una sólida reputación por su arquitectura probada, confiabilidad, integridad de datos, conjunto de características robustas, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para brindar soluciones innovadoras y de alto rendimiento de manera consistente (PostgreSQL Global Development Group, s.f.).

**3.2.18. *Amazon Web Services.*** Se trata de la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 175 servicios integrales de centros de datos a nivel global. Millones de clientes, incluyendo las empresas emergentes que crecen más rápido, las compañías más grandes y los organismos gubernamentales líderes, están utilizando AWS para reducir los costos, aumentar su agilidad e innovar de forma más rápida. AWS cuenta con una cantidad de servicios y de características incluidas en ellos que supera la de cualquier otro proveedor de la nube, ofreciendo desde tecnologías de infraestructura como cómputo, almacenamiento y bases de datos hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, lagos de datos y análisis e internet de las cosas (Amazon Web Services Inc, s.f.). Para este proyecto se emplearon varios de estos servicios.

**3.2.19. *Elastic beanstalk.*** Servicio fácil de utilizar para implementar y escalar servicios y aplicaciones web desarrollados con Java, .NET, PHP, Node.js, Python, Ruby, Go y

Docker en servidores familiares como Apache, Nginx, Passenger e IIS. Se trata de una plataforma como servicio que facilita el uso de otros servicios de AWS.

Elastic Beanstalk administra de manera automática la implementación, desde el aprovisionamiento de la capacidad, el equilibrio de carga y el escalado automático hasta la monitorización del estado de la aplicación. Al mismo tiempo, se tiene control de los recursos de Amazon Web Services que alimentan la aplicación y se puede acceder a los recursos subyacentes cuando se quiera (Amazon Web Services Inc, s.f.).

**3.2.20. *Elastic Compute Cloud (EC2)*.** Servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable. Está diseñado para simplificar el uso de la informática en la nube a escala web para los desarrolladores de aplicaciones. La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar capacidad con una fricción mínima. Proporciona un control completo sobre los recursos informáticos y puede ejecutarse en el entorno informático acreditado de Amazon (Amazon Web Services Inc, s.f.).

**3.2.21. *Relational Database Service (RDS)*.** Servicio para configurar, utilizar y escalar una base de datos relacional en la nube. El servicio suministra capacidad rentable y escalable al mismo tiempo que automatiza las arduas tareas administrativas, como el aprovisionamiento de hardware, la configuración de bases de datos, la implementación de parches y la creación de copias de seguridad. Libera a los desarrolladores web de estas tareas para que puedan concentrarse en sus aplicaciones y darles el rendimiento rápido, la alta disponibilidad, la seguridad y la compatibilidad que necesitan.

El servicio proporciona seis motores de bases de datos conocidos entre los que elegir, incluidos Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database y

SQL Server (Amazon Web Services Inc, s.f.).

**3.2.22. *Simple Storage Service (S3)*.** Servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en el sector. Cualquier cliente puede almacenar y proteger cualquier volumen de datos para los más variados fines, como usarlos en lagos de datos, sitios web, aplicaciones móviles, procesos de copia de seguridad y restauración, operaciones de archivado, aplicaciones empresariales, dispositivos IoT y análisis de big data. Amazon S3 proporciona características de administración fáciles de utilizar que le permiten organizar los datos y configurar sofisticados controles de acceso con objeto de satisfacer requisitos empresariales, organizativos y de conformidad. Amazon S3 está diseñado para ofrecer una durabilidad del 99,999999999 % (11 nueves) y almacena datos de millones de aplicaciones para empresas de todo el mundo (Amazon Web Services Inc, s.f.).

### **3.3. Marco de arquitectura software**

Se describe el estilo de la arquitectura de software usado para el desarrollo de la aplicación web.

**3.3.1. *Modelo-vista-controlador (MVC)*.** El modelo-vista-controlador es un patrón de software que se utiliza para el desarrollo de servicios web, como aplicaciones; independiente del lenguaje de programación. Es uno de los paradigmas más utilizados en el desarrollo web. La arquitectura comprende 3 componentes: el modelo, maneja los datos junto con sus reglas y lógica; la vista, se trata de la presentación de los datos a través de una interfaz de usuario gráfica, y el controlador; sirve como mediador entre

las dos anteriores al aceptar peticiones que se dan en la vista y dar órdenes al modelo para presentar datos como respuesta a la misma vista (Raji, 2017).

#### **4. Estado del arte**

Existen algunos ejemplos de sitios de reseñas donde estudiantes tienen la oportunidad de escribir sobre los profesores que ven y las instituciones educativas a las cuales asisten, pero las más populares y grandes están presentes más que todo en países angloparlantes. A continuación se describen las más usadas.

##### **4.1. Ratemyprofessors.com**

Desde su fundación en 1999 por John Swapceinski, RateMyProfessors ha crecido en tamaño y popularidad. Cualquier persona que entre al sitio puede buscar y leer reseñas de profesores contribuidas por estudiantes sin necesidad de crear una cuenta o iniciar sesión. En la página donde se pueden visualizar las reseñas sobre los profesores se puede encontrar información como número de reseñas, dificultad promedio, claridad promedio del profesor, calidad general, entre otras métricas. El sitio da acceso a más de 6 millones de reseñas y calificaciones de profesores escritas por estudiantes sobre más de un millón de profesores de universidades de los Estados Unidos, Canadá y el Reino Unido. Universidades y profesores de estos tres países son los únicos que se encuentran, limitando el alcance del sitio a esta parte del mundo anglosajón.

Usuarios que tomaron o estén tomando el curso de un profesor en particular pueden publicar una calificación y reseña sobre cualquier profesor que se encuentre listado en el sitio. Además, los usuarios pueden incluir profesores que no se encuentren en el sitio. Para poder

publicar una reseña un usuario debe calificar al profesor en una escala de 1-5 en las categorías de: calidad general y nivel de dificultad. El calificador tiene la opción de decir si le gustaría dar clases otra vez con el profesor, si la clase fue elegida por crédito, si la asistencia es obligatoria, si se usa libro, qué nota sacó el estudiante, e incluir un comentario de máximo 350 caracteres. También se puede seleccionar hasta 3 tags de una lista de 20 que describen el profesor (Crawford, s.f.).

Según la página de ayuda del sitio web, la calidad general del profesor debería representar que tan bien el profesor enseña el material del curso y que tan servicial es tanto dentro como fuera del salón de clases. La métrica de calidad general del profesor determina si su nombre será acompañado con una cara feliz (buena calidad), una cara triste (calidad pobre), o una cara sin expresión (calidad promedio) (Wikipedia, s.f.).

#### **4.2. Ratemyteachers.com**

Lanzada en 2001, RateMyTeachers permite calificar profesores e instituciones como colegios y universidades. Al escribir una reseña se pide calificar de 1 a 5 aspectos como facilidad, servicial, conocimiento y claridad, estos dos últimos usados para calcular la “calidad promedio” del profesor o institución. Actualmente RateMyTeachers sirve para los países: Estados Unidos, Reino Unido, Canadá, Irlanda, Australia y Nueva Zelanda. De acuerdo con el sitio su propósito es ayudar a responder una pregunta: “¿Qué es lo que necesito saber cómo estudiante para maximizar la probabilidad de éxito en alguna clase?” Para el año 2010 más de once millones de profesores habían sido calificados en el sitio.

Anteriormente el sitio estaba abierto al público sin registro ni requerimiento de revelar nombre o apellido al realizar reseñas. Esto llevó a algunos a preocuparse de que quienes usan el sitio puedan dejar comentarios difamatorios sin repercusiones. En 2007 el colegio de

maestros de Queensland pidió el cierre del sitio hasta que pueda ser establecido que provee un servicio creíble. En 2003 el sindicato de profesores Maestros Unidos del estado de Nueva York solicitó una orden judicial para cerrar el sitio, aunque la solicitud fue denegada por motivos de libertad de expresión.

Después de pasar a nuevos propietarios en 2018. Un análisis de la base de datos de comentarios por parte de la nueva administración encontró que había un enorme número de comentarios que violaban los términos y condiciones del sitio. También concluyeron que no había una forma efectiva de separar comentarios en cumplimiento de los comentarios en no cumplimiento de los términos y condiciones. Es por esto que, para cumplir con sus propios términos, el sitio quitó todos los comentarios escritos antes del periodo escolar 2019-2020. Esto provocó descontento en muchos profesores con comentarios positivos en el sitio. Sin embargo, al hacerlo el sitio cumplía con regulaciones mientras se mantenía como una útil plataforma para estudiantes (Wikipedia, s.f.).

### **4.3. DetodoUIS**

Es la aplicación móvil del sitio web [www.detodouis.com](http://www.detodouis.com), dedicada a tratar temas relacionados con la comunidad estudiantil de la Universidad Industrial de Santander. En la app se pueden encontrar funciones como: calendario, puntajes de corte, calcular valor del semestre, calculadora de promedio ponderado y ver materias de contexto, idiomas y cátedras, etc. La aplicación ha tenido más de cinco mil descargas en la tienda de aplicaciones móviles de Google (Google Play, s.f.).

La aplicación presenta entre sus características la posibilidad de dar y recibir referencias de profesores. Al buscar y seleccionar un profesor se puede visualizar la siguiente información: nombre completo del profesor, la escuela a la que pertenece, correo institucional, cantidad de

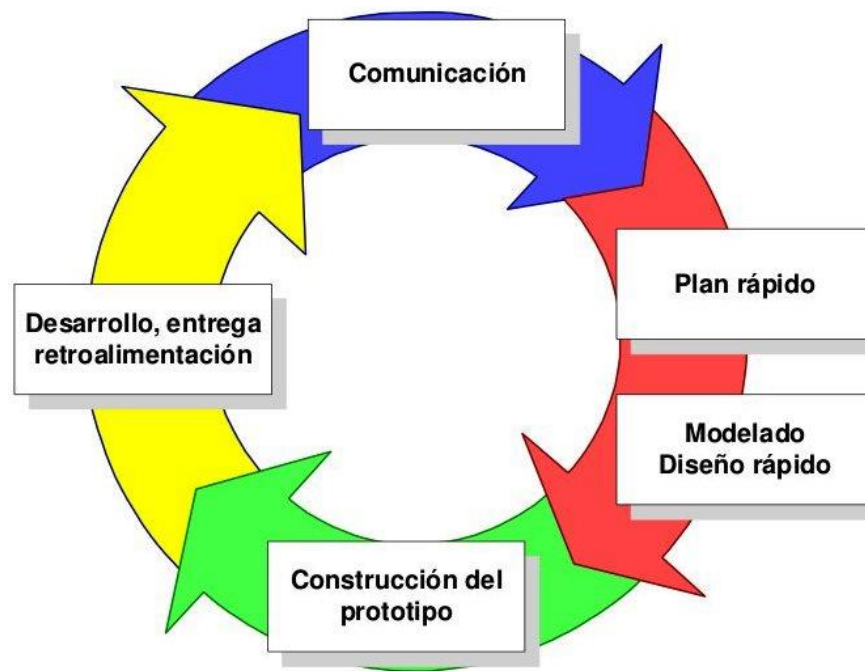
comentarios y puntaje promedio de las referencias. Debajo de esta información se pueden apreciar las referencias hechas por usuarios registrados de la aplicación. En las referencias se puede leer: nombre de usuario del calificador, puntaje del profesor (medido de 1 a 5 estrellas), fecha de publicación de la referencia y el comentario escrito por el calificador sobre el profesor. Para el comentario la aplicación invita al usuario a compartir alguna referencia, sugerencia, recomendación, comentario o consejo sobre el profesor seleccionado.

## **5. Metodología**

En esta parte se describe la metodología de desarrollo de software relacionada con el proyecto. Teniendo en cuenta el modelo de prototipos se prosiguieron con las siguientes fases: Análisis, requerimientos, diseño, implementación, pruebas, validación y salida a producción del prototipo o producto final.

La creación de prototipos es un gran enfoque para probar diferentes ideas y suposiciones, pero debe ser lo suficientemente inteligente como para saber sobre el modelo de creación de prototipos y cuánto tiempo y esfuerzos debe dedicarle. El modelo de prototipos es un método que ayuda con una estimación temprana del producto final o cualquier sistema de software cuando se desconoce cómo debería ser el producto final. Un prototipo funciona como prueba para probar todo el procedimiento. De esta muestra de prueba, aprendemos e intentamos construir un producto final superior (Mekhiya, 2016).

**Figura 1.** *Modelo de prototipos*



## 6. Desarrollo del proyecto

Se presenta el diseño del software donde se incluye: requisitos funcionales y no funcionales, casos de uso, diagramas de actividades, diseño de interfaz gráfica y modelo de base de datos.

### 6.1. Análisis y requerimientos

A continuación se listan y describen los requerimientos funcionales y requerimientos no funcionales de la aplicación, definidos al inicio y durante el desarrollo del proyecto.

Se llegó a los requerimientos finales con ayuda de la observación de otros sitios de reseñas como [ratemyprofessors.com](http://ratemyprofessors.com) y [ratemyteachers.com](http://ratemyteachers.com), junto con la observación de sitios de redes sociales más visitadas por la población estudiantil: Facebook, Twitter, YouTube, etc.

**6.1.1. Requerimientos funcionales.** En esta parte se describen aquellas funciones que el software debe tener, presentando en cada requerimiento su nombre, prioridad y su respectiva descripción.

**Tabla 1.** *Requerimiento funcional: registrar usuario*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF01
<b>Nombre</b>	Registrar usuario
<b>Prioridad</b>	Alta
<b>Descripción</b>	Los usuarios del sitio podrán registrarse ingresando los siguientes datos: nombre de usuario, correo electrónico y contraseña.

**Tabla 2.** *Requerimiento funcional: iniciar sesión*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF02
<b>Nombre</b>	Iniciar sesión
<b>Prioridad</b>	Alta
<b>Descripción</b>	Los usuarios registrados podrán iniciar sesión en el sitio usando los siguientes datos: correo electrónico y contraseña.

**Tabla 3.** *Requerimiento funcional: cerrar sesión*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF03
<b>Nombre</b>	Cerrar sesión
<b>Prioridad</b>	Alta
<b>Descripción</b>	Permitir a los usuarios cerrar sesión y eliminar el token que la identifica.

**Tabla 4.** *Requerimiento funcional: edición de cuenta de usuario*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF04
<b>Nombre</b>	Edición de cuenta de usuario
<b>Prioridad</b>	Media
<b>Descripción</b>	Los usuarios registrados podrán cambiar los datos con los cuales se registraron y cualquier otro almacenado: nombre de usuario, correo electrónico, contraseña, universidad a la que pertenece y escuela a la que pertenece.

**Tabla 5.** *Requerimiento funcional: recuperación de contraseña*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF05
<b>Nombre</b>	Recuperación de contraseña
<b>Prioridad</b>	Alta
<b>Descripción</b>	Los usuarios registrados podrán solicitar cambiar su contraseña si la han olvidado, haciendo uso de su correo electrónico para recibir un mensaje que los llevará al formulario para crear una nueva contraseña.

**Tabla 6.** *Requerimiento funcional: gestión de universidades*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF06
<b>Nombre</b>	Gestión de universidades desde cuenta administrativa
<b>Prioridad</b>	Alta
<b>Descripción</b>	Desde la cuenta administradora del sitio se podrá consultar, agregar, modificar y eliminar universidades de la base de datos.

**Tabla 7.** *Requerimiento funcional: gestión de escuelas*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF07
<b>Nombre</b>	Gestión de escuelas desde cuenta administrativa
<b>Prioridad</b>	Alta
<b>Descripción</b>	Desde la cuenta administradora del sitio se podrá consultar, agregar, modificar y eliminar escuelas de la base de datos.

**Tabla 8.** *Requerimiento funcional: gestión de profesores*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF08
<b>Nombre</b>	Gestión de profesores desde cuenta administrativa
<b>Prioridad</b>	Alta
<b>Descripción</b>	Desde la cuenta administradora del sitio se podrá consultar, agregar, modificar y eliminar profesores de la base de datos.

**Tabla 9.** *Requerimiento funcional: gestión de usuarios*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF09

<b>Nombre</b>	Gestión de usuarios desde cuenta administrativa
<b>Prioridad</b>	Alta
<b>Descripción</b>	Desde la cuenta administradora del sitio se podrá consultar y eliminar usuarios de la base de datos.

**Tabla 10.** *Requerimiento funcional: gestión de reseñas*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF10
<b>Nombre</b>	Gestión de reseñas
<b>Prioridad</b>	Alta
<b>Descripción</b>	<p>Los usuarios registrados podrán consultar, agregar, modificar y eliminar reseñas que hayan escrito sobre universidades, escuelas o profesores; dando datos como: texto de opinión sobre la universidad, escuela o profesor, calificación que dan y dificultad que percibieron en la entidad que evalúan</p> <p>También se podrá consultar, agregar, modificar y eliminar todas las reseñas del sitio desde la cuenta administrativa.</p>

**Tabla 11.** *Requerimiento funcional: guardar reseñas favoritas*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF11

<b>Nombre</b>	Guardar reseñas favoritas
<b>Prioridad</b>	Media
<b>Descripción</b>	Los usuarios registrados podrán guardar reseñas favoritas para poder ser consultadas por ellos cuando quieran.

**Tabla 12.** *Requerimiento funcional: compartir reseñas*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF12
<b>Nombre</b>	Compartir reseñas
<b>Prioridad</b>	Media
<b>Descripción</b>	Los visitantes y usuarios registrados del sitio podrán compartir reseñas en otras aplicaciones web.

**Tabla 13.** *Requerimiento funcional: recomendar reseña*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF13
<b>Nombre</b>	Botón de “Recomendar” en reseñas
<b>Prioridad</b>	Media
<b>Descripción</b>	Los usuarios registrados podrán recomendar reseñas a través de

	un botón de Recomendar. Al lado del botón se podrá ver cuantas recomendaciones ha recibido una reseña.
--	--

**Tabla 14.** *Requerimiento funcional: consultar reseñas recomendadas*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF14
<b>Nombre</b>	Consultar reseñas recomendadas
<b>Prioridad</b>	Media
<b>Descripción</b>	Los usuarios registrados podrán consultar todas las reseñas que han recomendado en el sitio.

**Tabla 15.** *Requerimiento funcional: suscripción a entidades*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF15
<b>Nombre</b>	Suscribirse a universidades, escuelas y profesores.
<b>Prioridad</b>	Alta
<b>Descripción</b>	Los usuarios registrados podrán suscribirse a las universidades, escuelas y profesores que estén en el sitio, para ser notificados cuando tengan una nueva reseña escrita por otro usuario.

**Tabla 16.** *Requerimiento funcional: ver notificaciones*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF16
<b>Nombre</b>	Ver notificación de nuevas reseñas
<b>Prioridad</b>	Alta
<b>Descripción</b>	Los usuarios registrados podrán consultar las notificaciones de nueva reseña escrita sobre las universidades, escuelas o profesores a los que estén suscritos.

**Tabla 17.** *Requerimiento funcional: búsqueda de entidades*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF17
<b>Nombre</b>	Filtrar búsqueda de profesores, escuelas y universidades
<b>Prioridad</b>	Alta
<b>Descripción</b>	El buscador del sitio podrá ser usado para encontrar universidades, escuelas y profesores que se encuentren en la base de datos.

**Tabla 18.** *Requerimiento funcional: visualizar reseñas de entidades*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF18
<b>Nombre</b>	Visualizar reseñas de universidades, escuelas y profesores
<b>Prioridad</b>	Alta
<b>Descripción</b>	Los visitantes y usuarios registrados del sitio podrán consultar las reseñas escritas sobre universidades, escuelas y profesores, junto con los datos más relevantes de las entidades como: nombre, número de reseñas, número de suscriptores, calificación y dificultad promedio recibida en las reseñas.

**Tabla 19.** *Requerimiento funcional: visualizar reseñas más recomendadas*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF19
<b>Nombre</b>	Visualizar reseñas más recomendadas en la página principal.
<b>Prioridad</b>	Media
<b>Descripción</b>	Los visitantes y usuarios del sitio podrán ver las últimas y más recomendadas reseñas escritas en el sitio sobre universidades, escuelas y profesores.

**Tabla 20.** *Requerimiento funcional: visualizar últimas escuelas agregadas*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF20
<b>Nombre</b>	Visualizar las últimas escuelas agregadas en la página principal.
<b>Prioridad</b>	Media
<b>Descripción</b>	Los visitantes y usuarios registrados podrán visualizar desde la página principal las últimas escuelas que se hayan agregado al sitio.

**Tabla 21.** *Requerimiento funcional: ver profesores probables*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF21
<b>Nombre</b>	Ver profesores probables
<b>Prioridad</b>	Media
<b>Descripción</b>	Los usuarios registrados verán una lista de profesores que probablemente conocen o conocerán, después que el usuario haya especificado su universidad y escuela.

**Tabla 22.** *Requerimiento funcional: reportar reseña*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF22

<b>Nombre</b>	Reportar reseña
<b>Prioridad</b>	Alta
<b>Descripción</b>	Los usuarios registrados podrán reportar reseñas que consideren violan los términos de servicio del sitio. A su vez el administrador del sitio podrá ver los reportes y decidir si eliminar o no una reseña.

**Tabla 23.** *Requerimiento funcional: feed de usuario*

<b>Requerimiento funcional</b>	
<b>Requerimiento</b>	RF23
<b>Nombre</b>	Feed de usuario
<b>Prioridad</b>	Alta
<b>Descripción</b>	Los usuarios registrados, después de seleccionar la universidad y escuela a la que pertenecen, podrán ver las mejores reseñas que se escriban sobre su universidad, su escuela y los profesores de su escuela.

**6.1.2. Requerimientos no funcionales.** A continuación se describen aspectos no pertinentes con el comportamiento funcional del sistema, pero aun así deben ser visibles por el usuario.

**Tabla 24.** *Requerimiento no funcional: interfaz gráfica responsiva*

<b>Requerimiento no funcional</b>	
<b>Requerimiento</b>	RNF01
<b>Nombre</b>	Diseñar una interfaz gráfica responsiva.
<b>Prioridad</b>	Alta
<b>Descripción</b>	Los componentes del sitio se adaptarán a la pantalla desde la cual se presenten; ya sea computadora de escritorio, computador portátil, tablet o celular.

**Tabla 25.** *Requerimiento no funcional: interfaz gráfica SPA*

<b>Requerimiento no funcional</b>	
<b>Requerimiento</b>	RNF02
<b>Nombre</b>	Interfaz gráfica como SPA.
<b>Prioridad</b>	Alta
<b>Descripción</b>	La interfaz gráfica de usuario del sitio será una aplicación de una sola página creada con ayuda de ReactJS

**Tabla 26.** *Requerimiento no funcional: compatibilidad con navegadores*

<b>Requerimiento no funcional</b>	
<b>Requerimiento</b>	RNF03

<b>Nombre</b>	Compatibilidad con diferentes navegadores web
<b>Prioridad</b>	Alta
<b>Descripción</b>	La aplicación web deberá ser compatible con los más usados navegadores web.

## 6.2. Diseño del software

El diseño del software consiste en diferentes etapas anteriores a la creación del software, que ayuda a modelar el sistema completo.

**6.2.1. Casos de uso.** Con los casos de uso se busca visualizar las formas de usar un sistema por parte de diferentes actores involucrados. A continuación se pueden apreciar los casos de uso de la aplicación.

**Figura 2.** Casos de uso



Para un mejor entendimiento de los casos de uso a continuación se listan los actores que hacen parte del software, junto con aquellos requerimientos funcionales que cumplen y una breve descripción de su participación en el sistema

**Actores del software:**

- Visitante
- Usuario registrado
- Administrador

**Tabla 27.** Descripción de actor visitante

<b>Actor</b>	Visitante
<b>Requerimientos funcionales</b>	RF12: Compartir reseñas
	RF17: Filtrar búsqueda de profesores, escuelas y universidades
	RF18: Visualizar reseñas de universidades, escuelas y profesores
	RF19: Visualizar reseñas más recomendadas en la página principal.
	RF20: Visualizar las últimas escuelas agregadas en la página principal.
<b>Descripción</b>	Usuario no registrado en el sitio, que puede realizar acciones que no requieren inicio de sesión.
<b>Tipo</b>	Secundario

**Tabla 28.** Descripción de actor usuario registrado

<b>Actor</b>	Usuario registrado
<b>Requerimientos funcionales</b>	RF01: Registrar usuario
	RF02: Iniciar sesión
	RF03: Cerrar sesión
	RF04: Edición de cuenta de usuario
	RF05: Recuperación de contraseña
	RF10: Gestión de reseñas
	RF11: Guardar reseñas favoritas
	RF12: Compartir reseñas
	RF13: Botón de “Recomendar” en reseñas
	RF14: Consultar reseñas recomendadas
	RF15: Suscribirse a universidades, escuelas y profesores.
	RF16: Ver notificación de nuevas reseñas
	RF17: Filtrar búsqueda de profesores, escuelas y universidades
	RF18: Visualizar reseñas de universidades, escuelas y profesores

	RF19: Visualizar reseñas más recomendadas en la página principal
	RF20: Visualizar las últimas escuelas agregadas en la página principal
	RF21: Ver profesores probables
	RF22: Reportar reseña
	RF23: Feed de usuario
<b>Descripción</b>	Usuario registrado en el sitio, que puede realizar acciones que requieren iniciar sesión en la aplicación web, junto con todas las acciones que realiza un visitante.
<b>Tipo</b>	Primario

**Tabla 29.** Descripción de actor administrador

<b>Actor</b>	Administrador
<b>Requerimientos</b>	RF06: Gestión de universidades desde cuenta administrativa
	RF07:Gestión de escuelas desde cuenta administrativa
	RF08:Gestión de profesores desde cuenta administrativa
	RF09: Gestión de usuarios desde cuenta administrativa
	RF10: Gestión de reseñas

<b>funcionales</b>	RF22: Reportar reseña
<b>Descripción</b>	Usuario administrador del sitio, el único que puede realizar acciones como: gestionar universidades, escuelas, profesores y reseñas de cualquier usuario. Así como eliminar usuarios registrados y ver reportes. Todo desde el panel de administrador de Django.
<b>Tipo</b>	Primario

**6.2.2. Diseño de la interfaz gráfica.** El objetivo de los siguientes esquemas es modelar como debe ser la interfaz gráfica de la aplicación web. Esto facilita enormemente su posterior construcción haciendo uso de los componentes de React y los estilos de la librería Bulma. Los siguientes esquemas fueron diseñados haciendo uso de la herramienta web [figma.com](https://www.figma.com)

La observación de sitios de redes sociales más visitadas por la población estudiantil como Facebook, Twitter, YouTube sirvió de inspiración para muchos aspectos del diseño presentado a continuación.

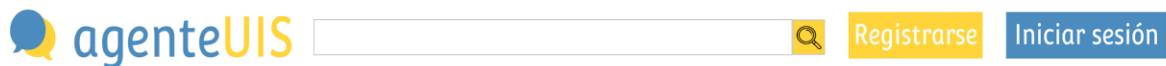
**Diseño del logo y nombre del sitio:** Tanto el logo como el título usan los 2 colores principales que se verán en la aplicación.

**Figura 3.** *Diseño del logo*



**Diseño de la cabecera:** se diseñó una cabecera principal que se verá en cada página del sitio. Está formada por tres simples componentes: logo, buscador y botones de la cuenta.

**Figura 4.** *Diseño de cabecera*



**Diseño del menú principal:** se optó por un menú lateral que sería usado por la mayoría de componentes principales y direcciona a los sitios más importantes de la aplicación.

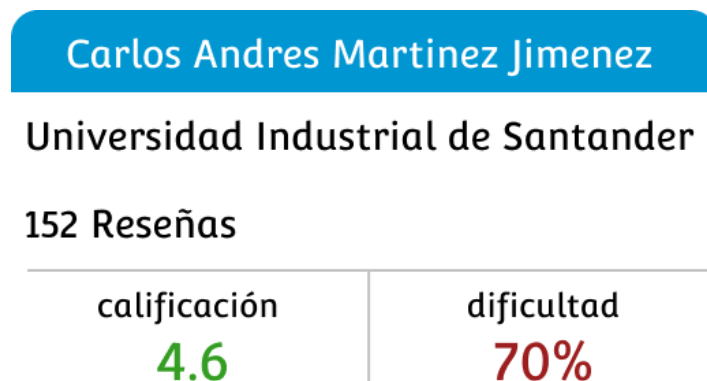
**Figura 5.** *Diseño de menú principal*



**Diseño de componentes reusables:** Los siguientes componentes se usan en varios lugares de la aplicación o hacen referencia a elementos que se repiten dentro de una lista. Para ellos se usa un estilo con los colores principales de la aplicación y fondos grises claros con letras en gris fuerte.

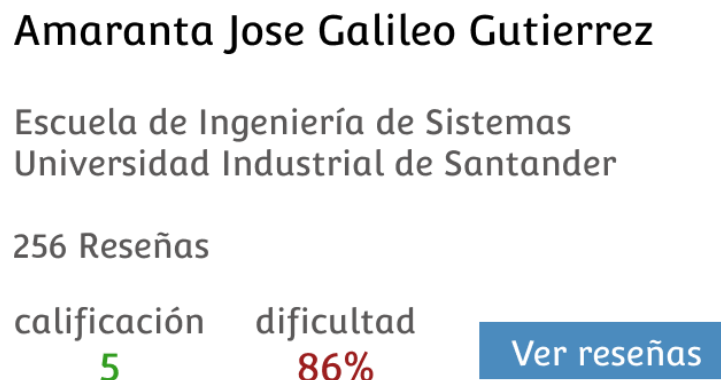
- Profesor en menú principal:** Diseñado para los profesores cuando son listados desde el menú lateral, muestra su información más relevantes como nombre, apellido, universidad donde trabaja, número de reseñas que ha recibido, calificación y dificultad promedio según las reseñas.

**Figura 6.** *Diseño componente profesor en menú principal*



- Profesor en lista:** Se diseñó un componente cuando los profesores son listados desde el perfil de una universidad o escuela, y cuando están entre resultados de búsqueda.

**Figura 7.** *Diseño de componente profesor en lista*



- **Detalles de profesor:** Vista principal del profesor donde se presentan todos sus datos: nombre completo, universidad y escuela donde trabaja, reseñas recibidas, número de suscriptores, calificación y dificultad promedio recibida por las reseñas.

**Figura 8.** *Diseño de componente detalles de profesor*

**Amaranta Jose Galileo Gutierrez**

Escuela de Ingeniería de Sistemas  
Universidad Industrial de Santander



256 Reseñas

calificación  
general

**5/5**

dificultad  
promedio

**86%**

Crear reseña

16 usuarios esperan nueva reseña

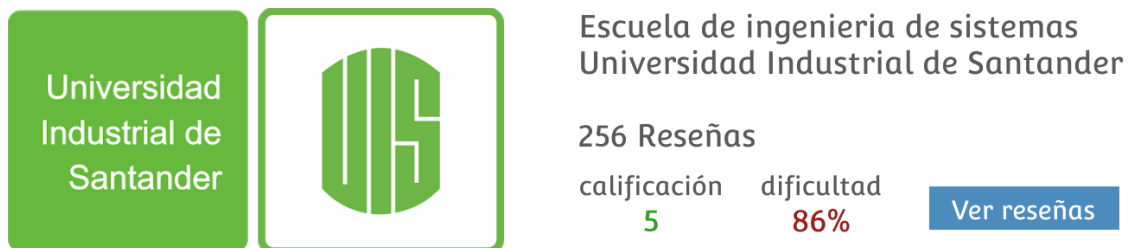
- **Institución en menú principal:** Se diseñó un componente para mostrar las instituciones (universidades y escuelas) desde el menú principal. Se muestra una miniatura del logo de la institución junto con su nombre, cantidad de reseñas, calificación y dificultad promedio.

**Figura 9.** *Diseño de componente detalles de universidad*



- **Escuela en lista:** Se diseñó un componente similar a profesor en lista cuando las escuelas están listadas desde el perfil de una universidad y cuando están entre resultados de búsqueda.

**Figura 10.** *Diseño de componente escuela en lista*



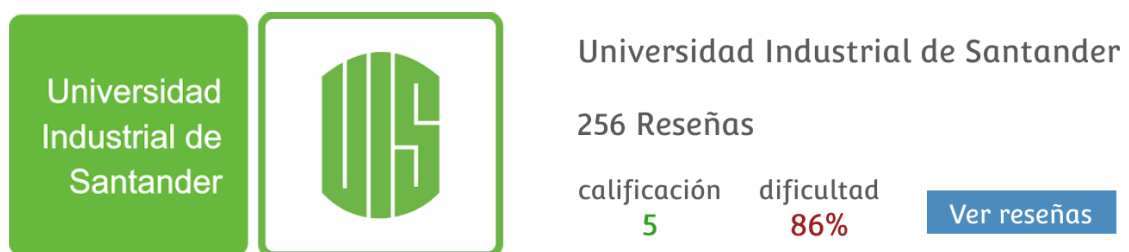
- **Detalles de escuela:** Componente principal para las escuelas donde se presentan todos sus datos: logo, nombre, universidad, reseñas recibidas, número de suscriptores, calificación y dificultad promedio recibida por las reseñas.

**Figura 11.** *Diseño de componente detalles de escuela*



- **Universidad en lista:** Diseño y presentación similar a escuelas en lista, usado cuando se encuentra entre resultados de búsqueda.

**Figura 12.** *Diseño de componente universidad en lista*



- **Detalles de universidad:** Diseño y presentación similar a los detalles de escuela, presenta los datos de una universidad.

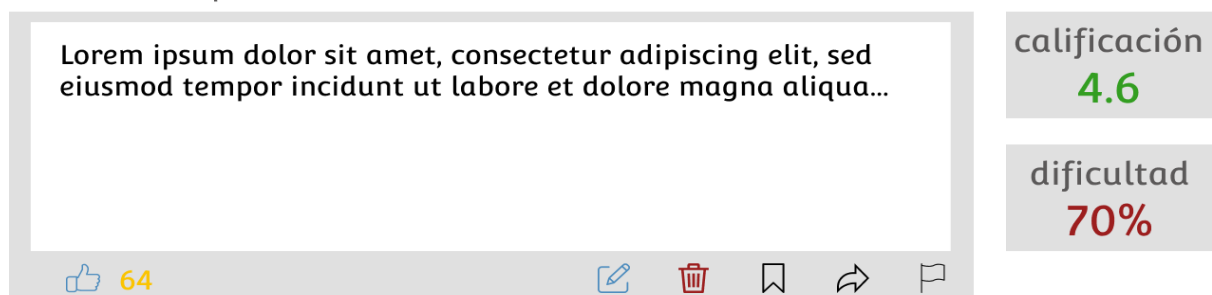
**Figura 13.** *Diseño de componente detalles de universidad*



- **Reseña:** Se diseñó la forma principal de las reseñas que se verán en los perfiles de universidades, escuelas y profesores, de tal forma que se visualiza muy bien la información más relevante de la reseña; el texto, la calificación y la dificultad dada, y las demás opciones quedan como secundarias a la vista; cantidad de recomendaciones recibidas, botón de editar, borrar, guardar, compartir, reportar.

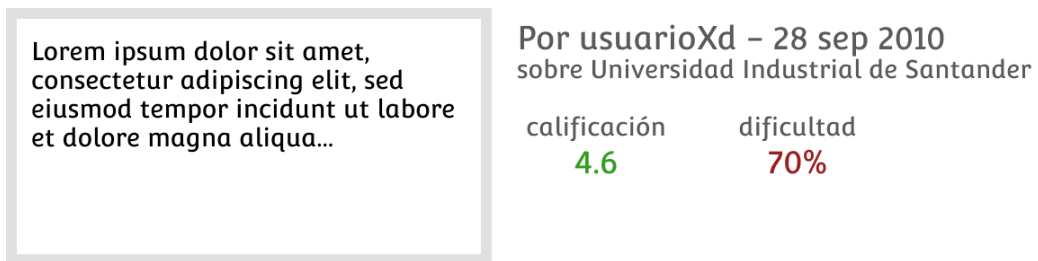
**Figura 14.** *Diseño de componente reseña*

usuario – 28 sep 2010



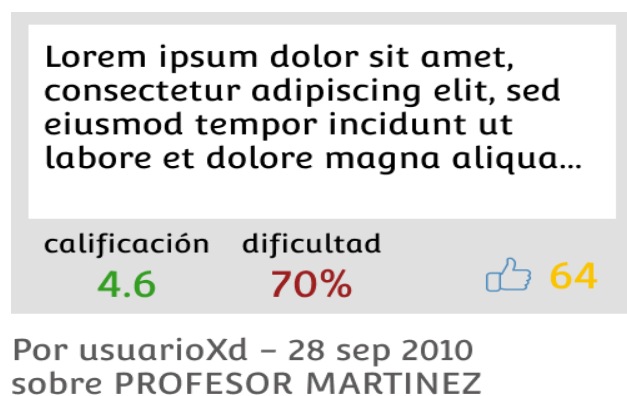
- **Reseña listada en resultados de búsqueda:** Se diseñó un componente cuando las reseñas hacen parte de los resultados de alguna búsqueda realizada. Se muestra la información más relevante de la reseña y se dejan de lado los demás botones.

**Figura 15.** *Diseño de componente reseña en resultados de búsqueda*



- **Reseña listada en menú principal:** Diseño de reseña usado para las reseñas que se presentan en la página principal del sitio. Se intenta seguir el mismo patrón de los anteriores diseños manteniendo sus colores y estilo, junto con su información más relevante.

**Figura 16.** *Diseño de componente reseña en menú principal*



- **Lista de profesores en panel derecho:** Se diseñó una lista con bordes redondeados y un color similar al primario para mostrar los nombres de los profesores que el usuario quizás conozca.

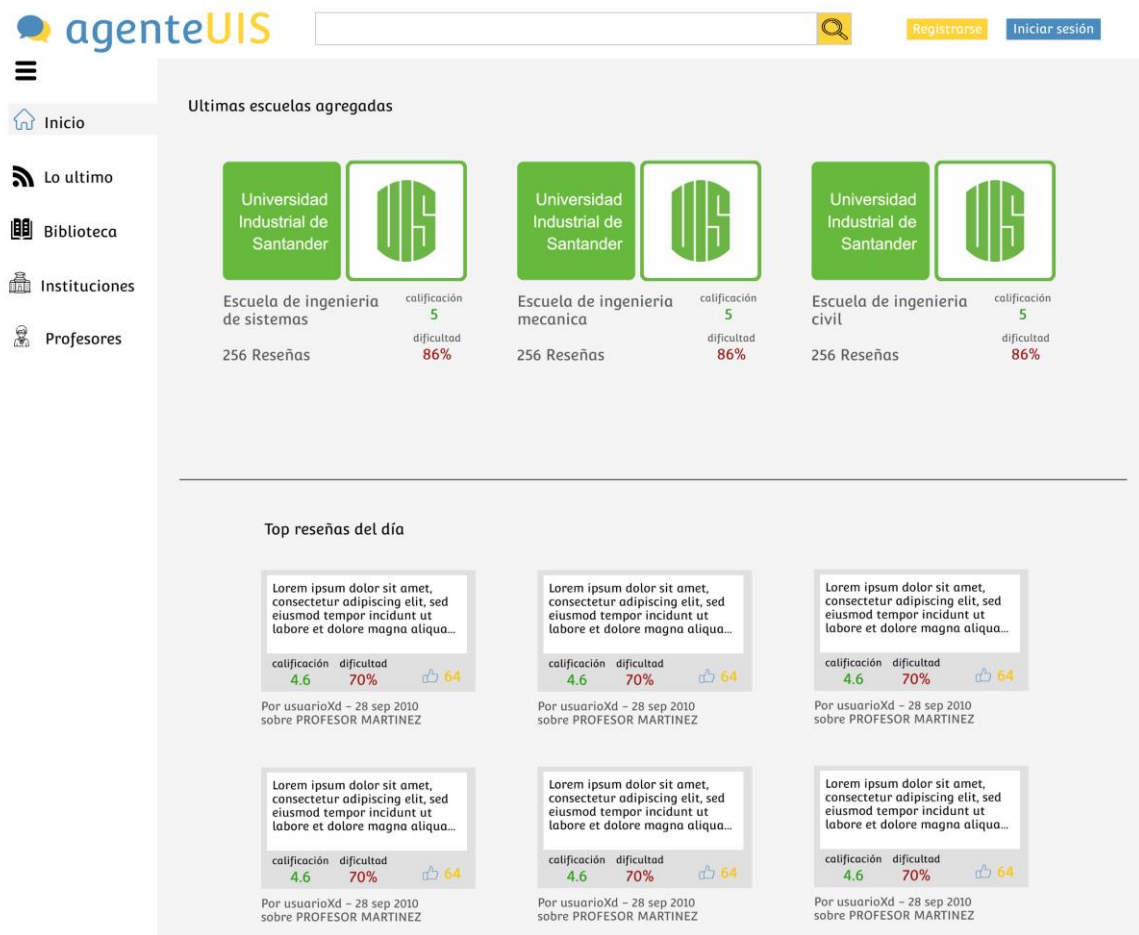
**Figura 17.** *Diseño de componente de profesores probables*



**Diseño de los componentes principales del contenido:** estos componentes comprenden aquellos que hacen uso de los componentes más pequeños para crear las páginas que compondrán la aplicación.

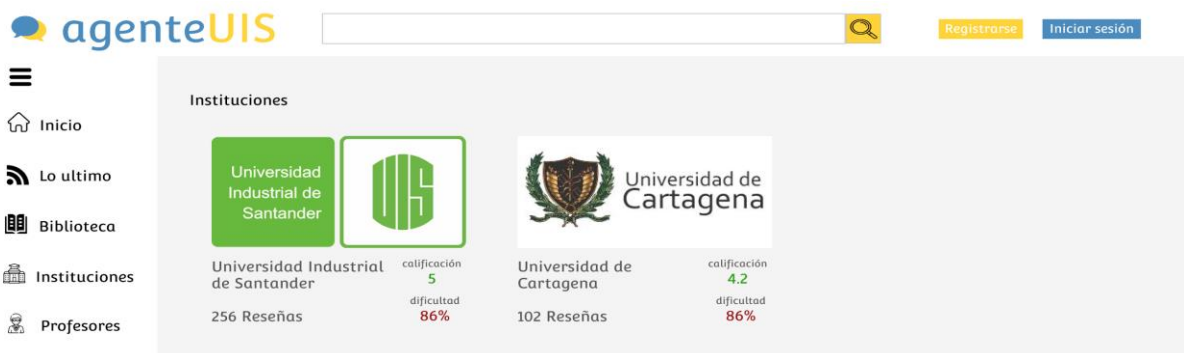
- **Página principal:** se diseñó una página principal donde aterrizan los usuarios al entrar al sitio. En esta se usan los componentes de la institución y reseña en menú principal para mostrar las últimas escuelas agregadas y las mejores reseñas respectivamente.

Figura 18. Diseño de página principal



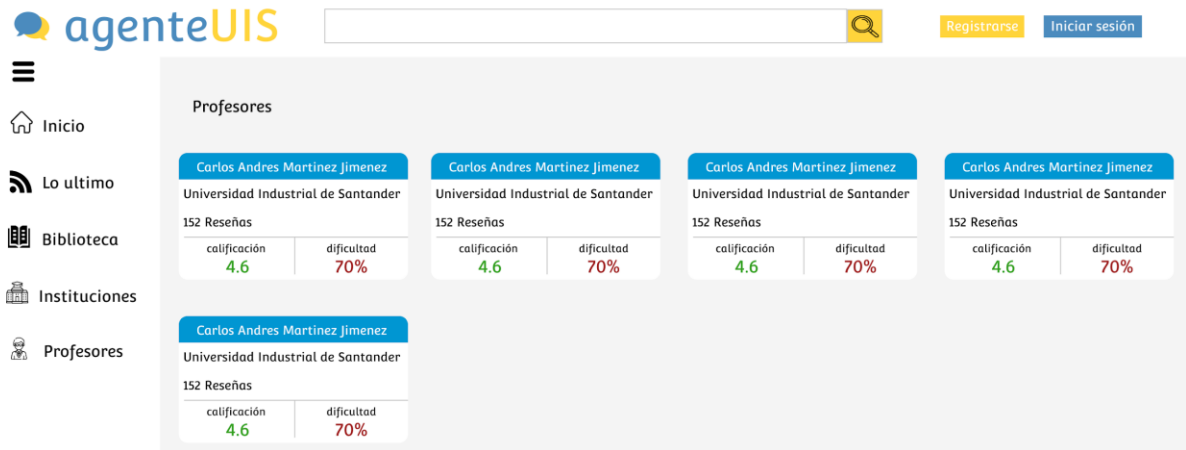
- **Lista de instituciones:** Se diseñó una lista de 3 columnas para presentar todas las instituciones que hay en el sitio, usando el componente de instituciones en menú principal.

Figura 19. Diseño de lista de instituciones



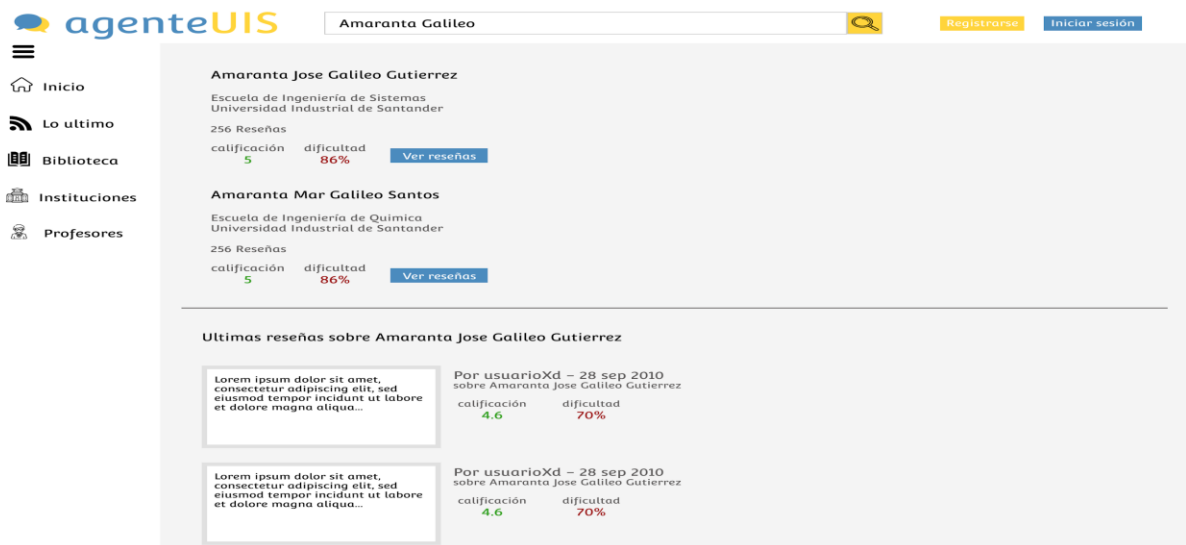
- **Lista de profesores:** Se diseñó una lista de 4 columnas usando el componente de profesores en menú principal para presentar los profesores del sitio.

Figura 20. Diseño de lista de profesores

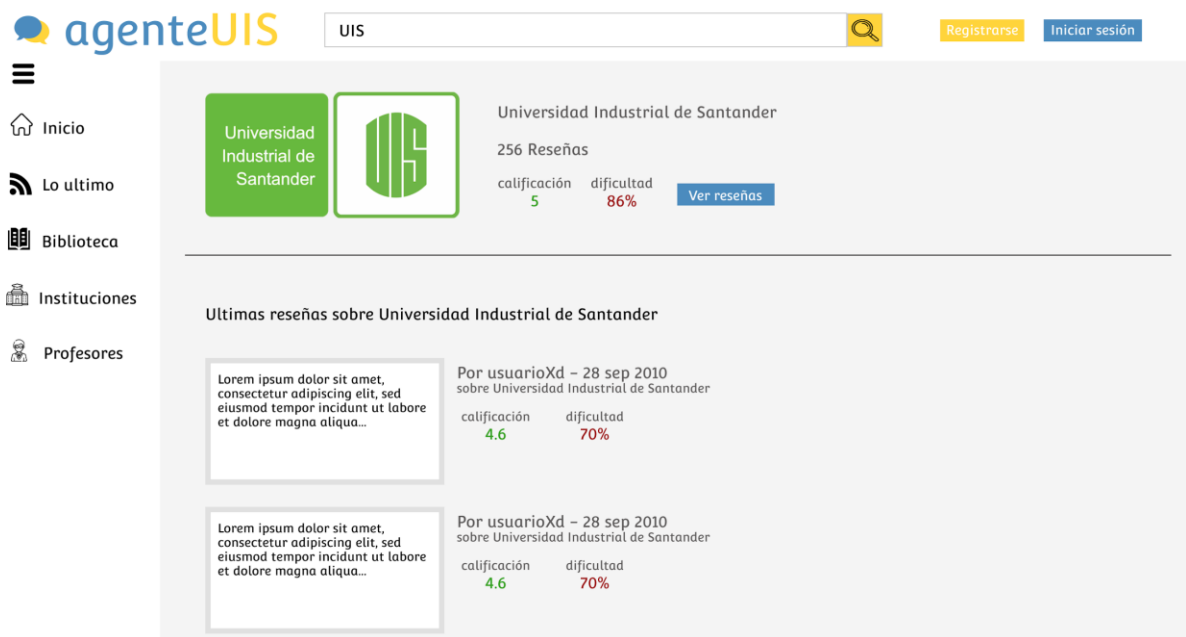


- **Resultados de búsqueda:** Se diseñó una lista de una columna para presentar los resultados de búsqueda para no cansar la vista intentando encontrar una entidad de entre varios elementos.

Figura 21. Diseño de resultados de búsqueda de profesores



**Figura 22.** *Diseño de resultados de búsqueda de universidades*



- **Reseñas de entidad o perfil de entidad:** Se diseñó una vista estilo perfil para presentar los detalles de las entidades (universidad, profesor o escuela) junto con una lista de las reseñas que han escrito sobre ellas.

**Figura 23.** *Diseño de perfil de instituciones*

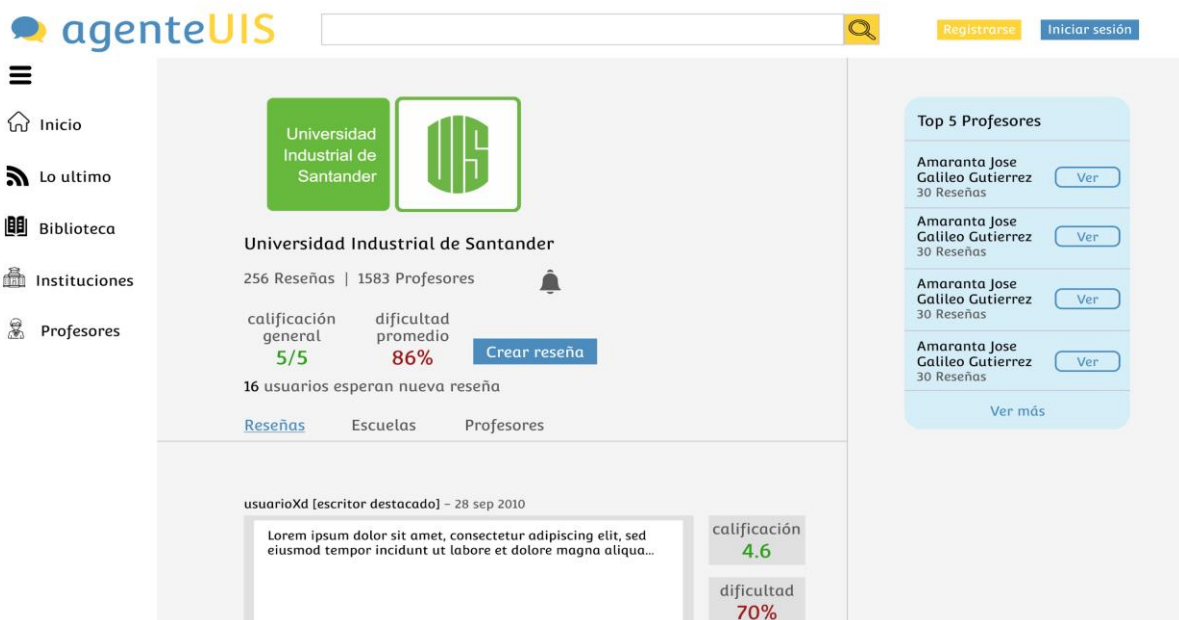
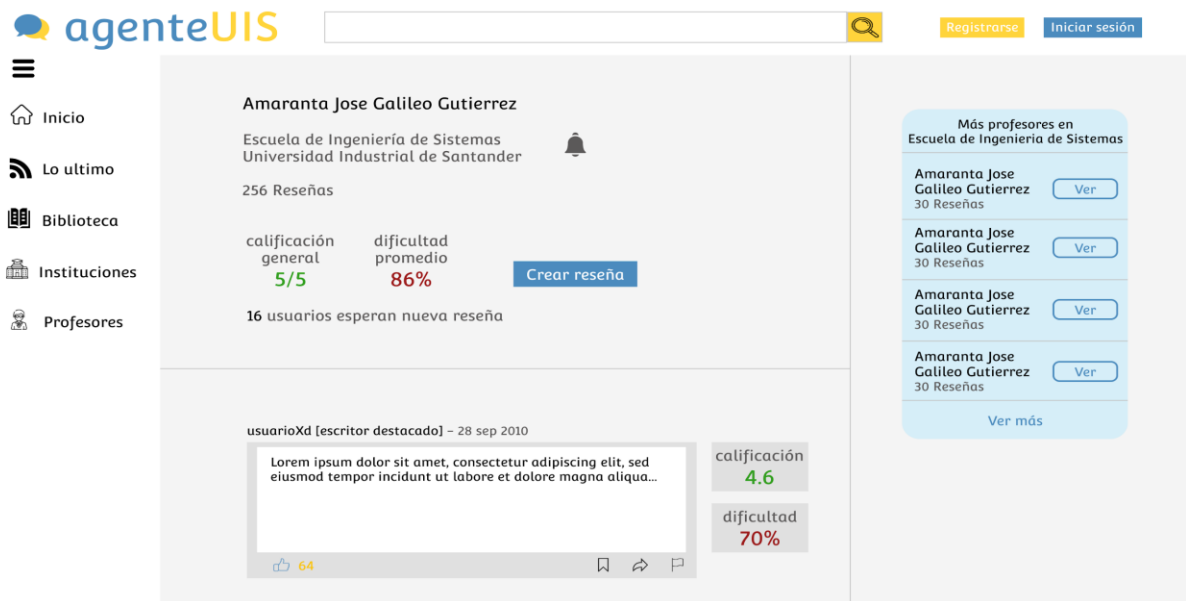


Figura 24. Diseño de perfil de profesores



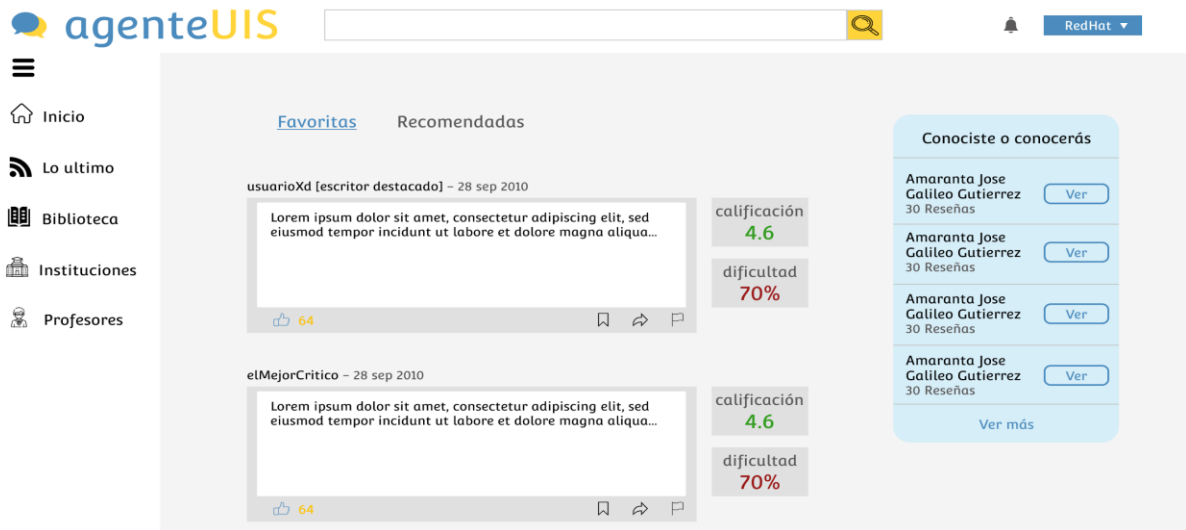
- **Formulario de creación de reseña:** Se diseñó una página aparte para la creación de reseñas para que sea más fácil presentar todos los campos requeridos y recomendaciones al usuario antes de enviar su reseña.

Figura 25. Diseño de formulario creación de reseñas



- **Biblioteca de usuario:** Se diseñó una vista para listar en una sola columna todas las reseñas que el usuario ha guardado o ha recomendado.

Figura 26. Diseño de biblioteca de usuario



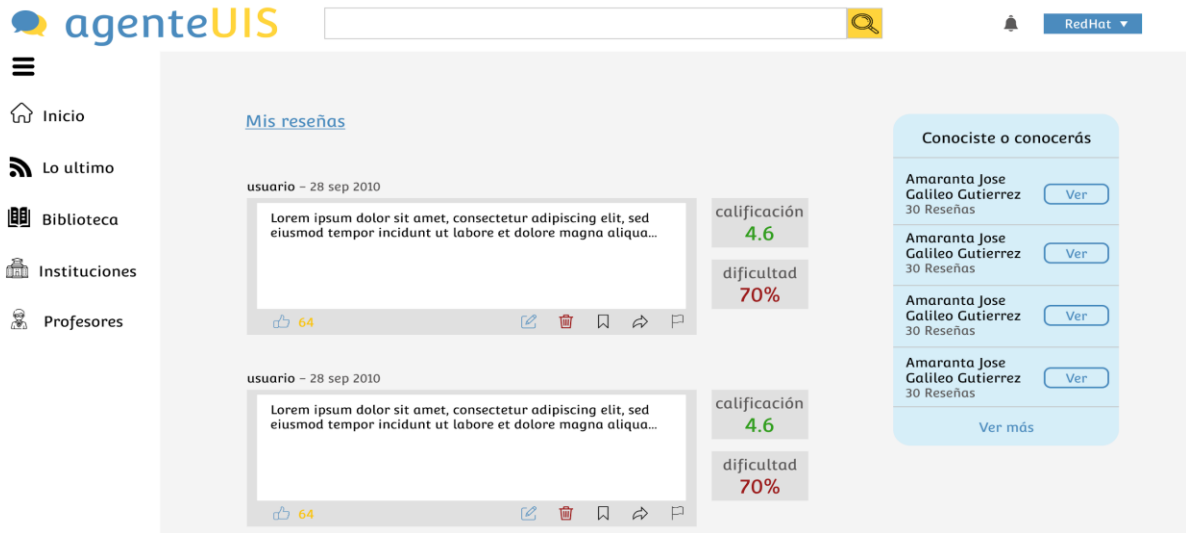
- **Feed de usuario:** Respetando el estilo para listar la biblioteca de usuario se diseñó la página para el feed de usuario.

Figura 27. Diseño de feed usuario



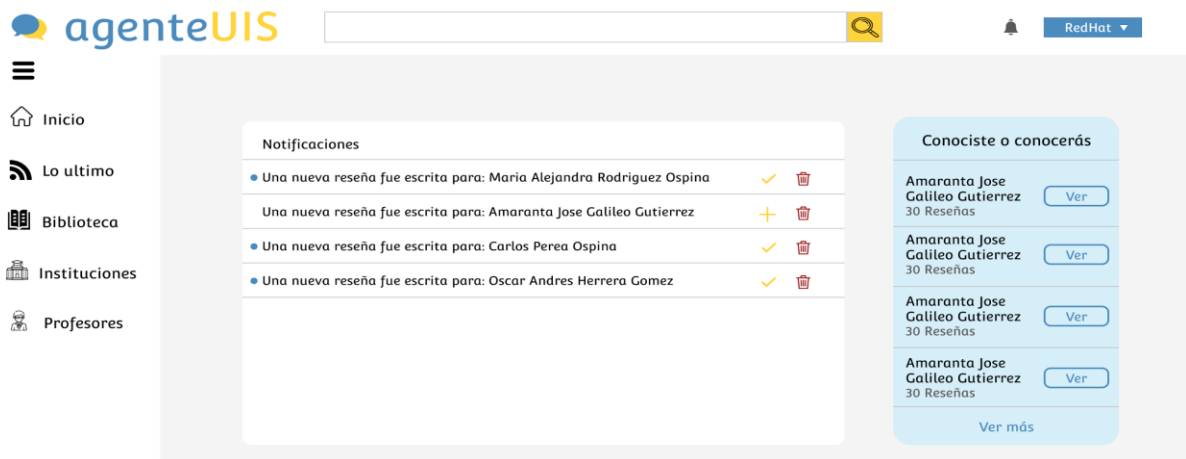
- **Reseñas de usuario:** Siguiendo el mismo patrón anterior se presentan las reseñas que han sido creadas por el usuario.

Figura 28. Diseño de reseñas de usuario



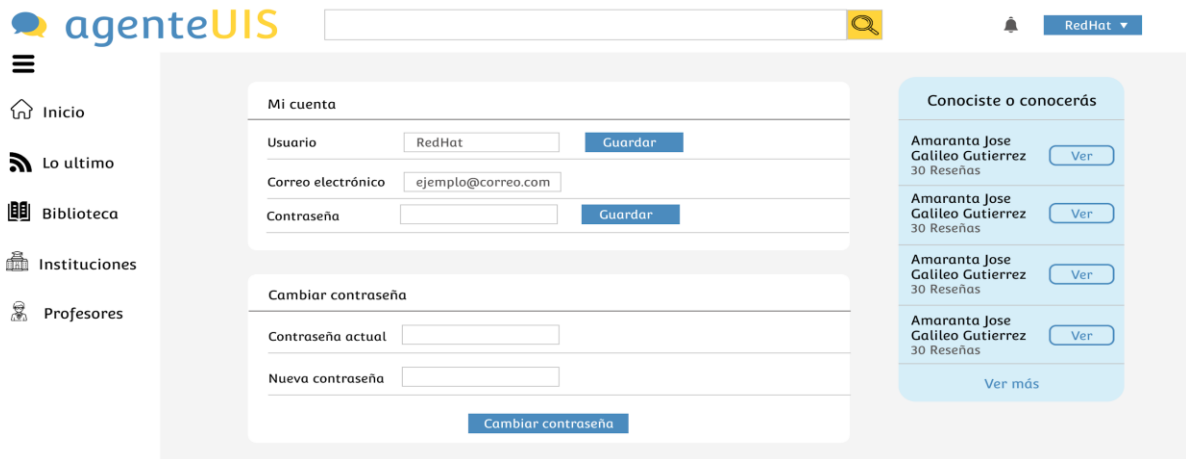
- **Notificaciones de usuario:** Se diseñó una caja con fondo blanco y bordes redondeados para listar las notificaciones recibidas por el usuario, junto con sus opciones de marcar como vista y eliminar.

Figura 29. Diseño de notificación de usuario



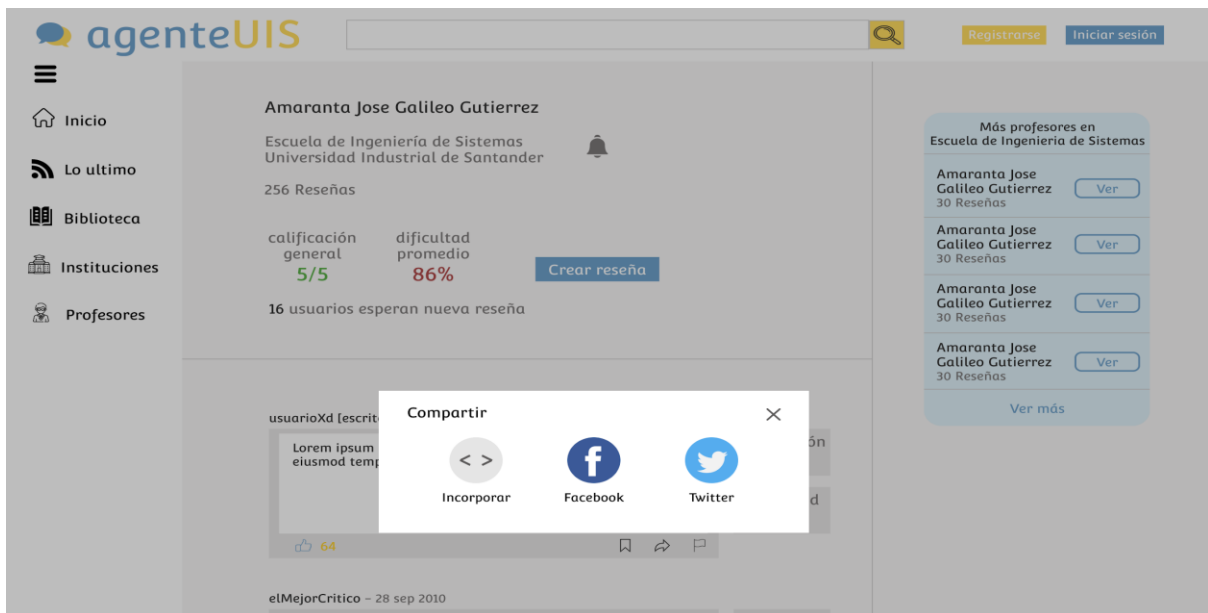
- **Edición de datos de usuario:** Se diseñaron cajas similares a las notificaciones, siguiendo el mismo patrón, para presentar los formularios de cambio de datos de usuario.

Figura 30. Diseño de edición de datos de usuario



- **Compartir reseña:** Se diseñó un modal para elegir entre las diferentes formas para compartir una reseña.

Figura 31. Diseño de compartir reseña



- **Formulario de registro:** Se diseñó una página completa para presentar el formulario de registro de usuario, respetando los colores principales del sitio para su botón de envío.

**Figura 32.** *Diseño de formulario de registro*

agenteUIS

Registrarse Iniciar sesión

**Registrarse en agenteUIS**

nombre de usuario

correo

contraseña

Registrarse

Regístrate y:

- Califica a tu profesores.
- Se notificado de nuevas reseñas sobre profesores que especifiques.
- Califica reseñas de otros usuarios.

- **Formulario de inicio de sesión:** Se diseñó una página completa para presentar el formulario de inicio de sesión manteniendo el mismo estilo del formulario de registro con los colores principales.

**Figura 33.** *Diseño de inicio de sesión*

agenteUIS

Registrarse Iniciar sesión

**Iniciar sesión en agenteUIS**

usuario

contraseña

Iniciar sesión

¿Has olvidado la contraseña?

6.2.3. *Diagramas de actividades* Se presentan los diagramas de actividades de las principales funcionalidades y a su vez las principales acciones de los actores de la aplicación.

**Figura 34.** Diagrama de actividades para administración de entidades

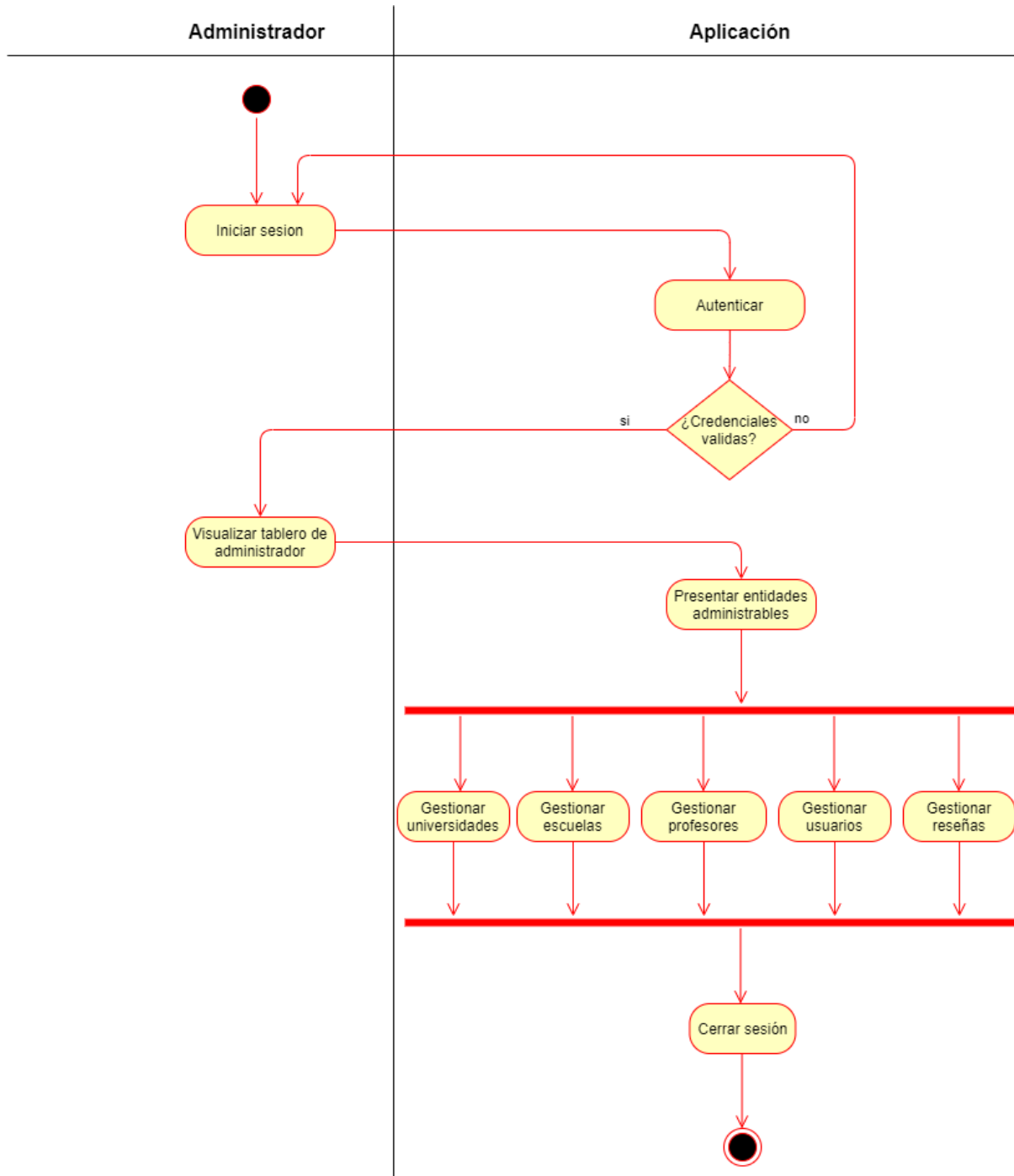


Figura 35. Diagrama de actividades para gestión de reseñas

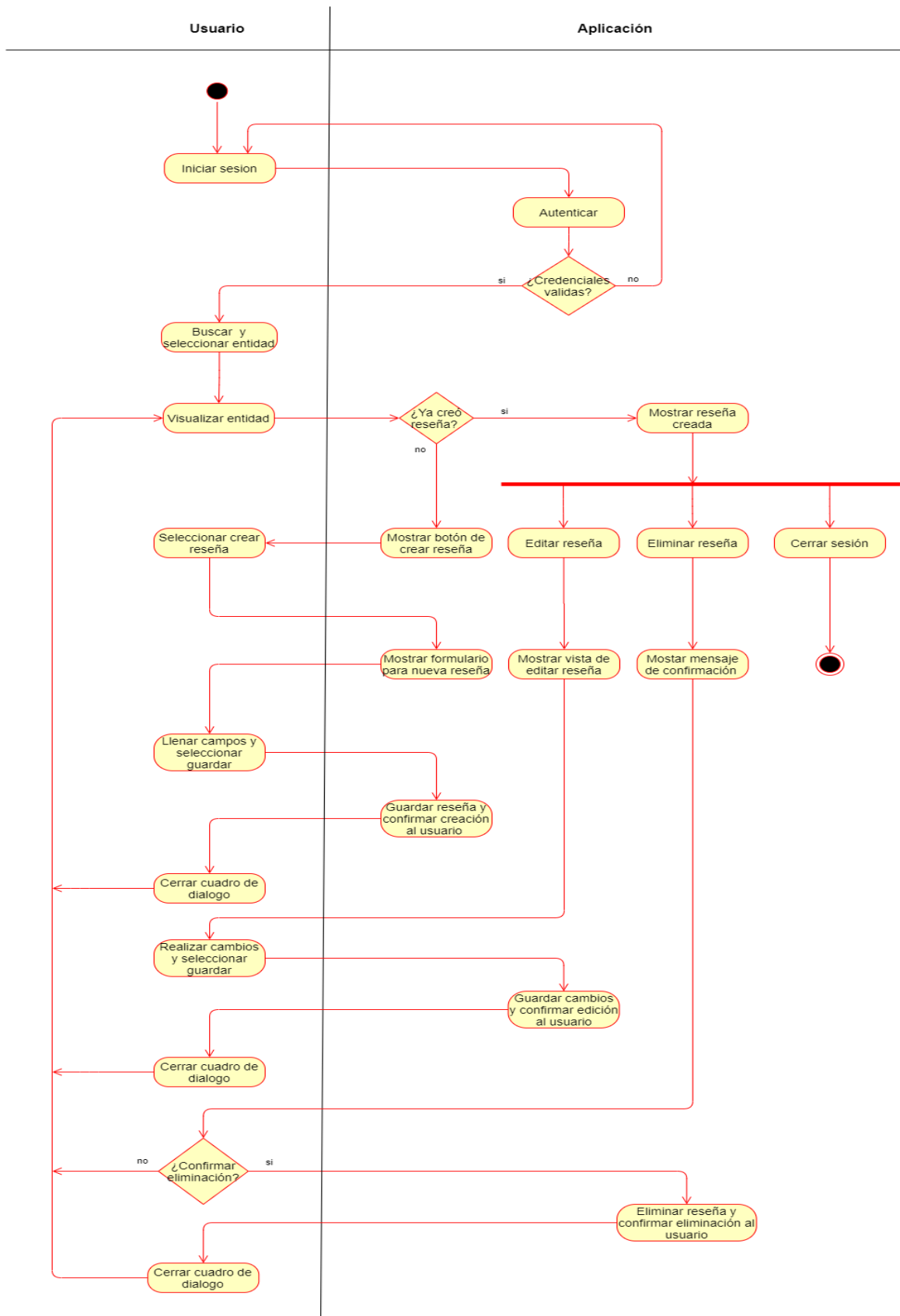
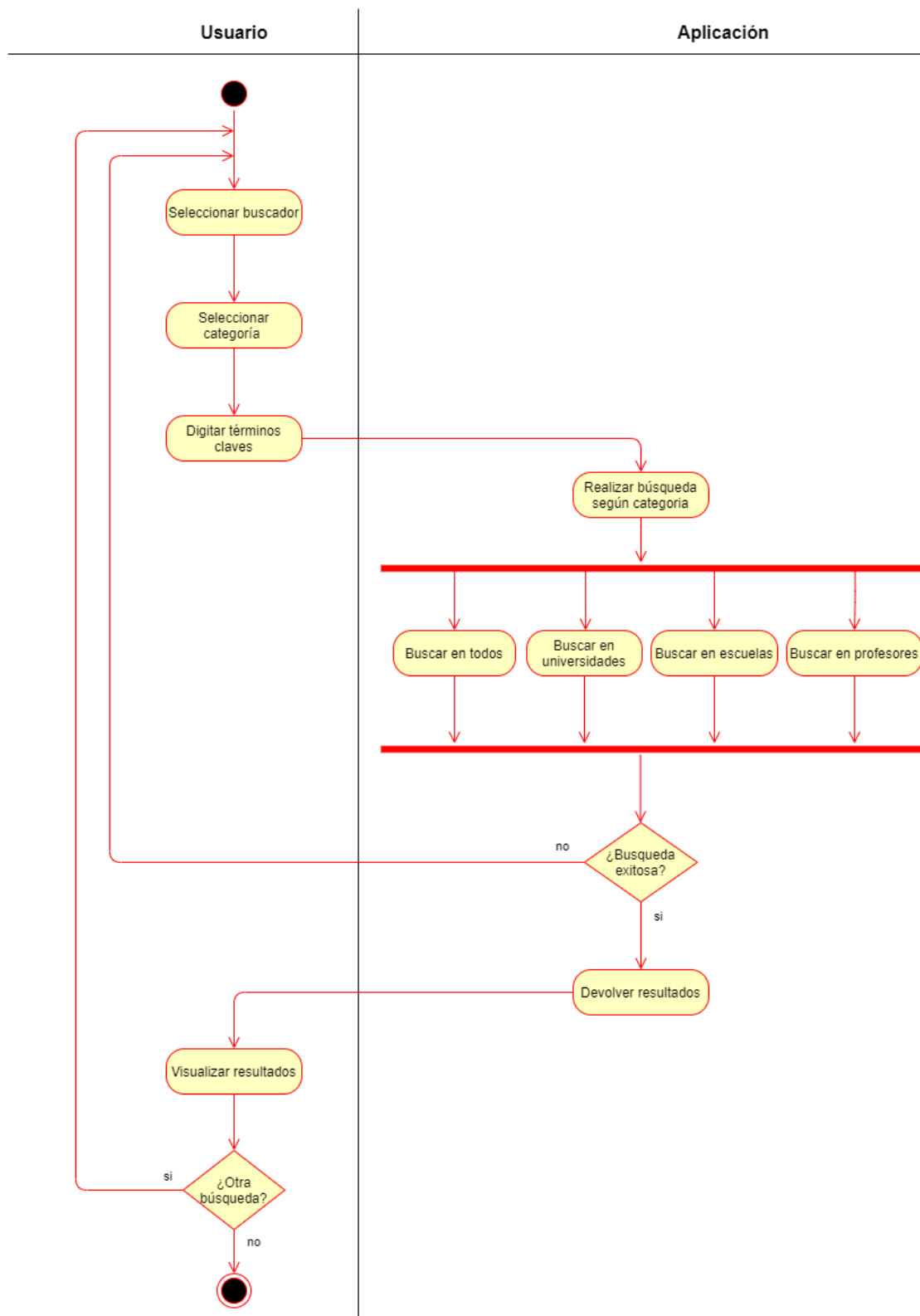
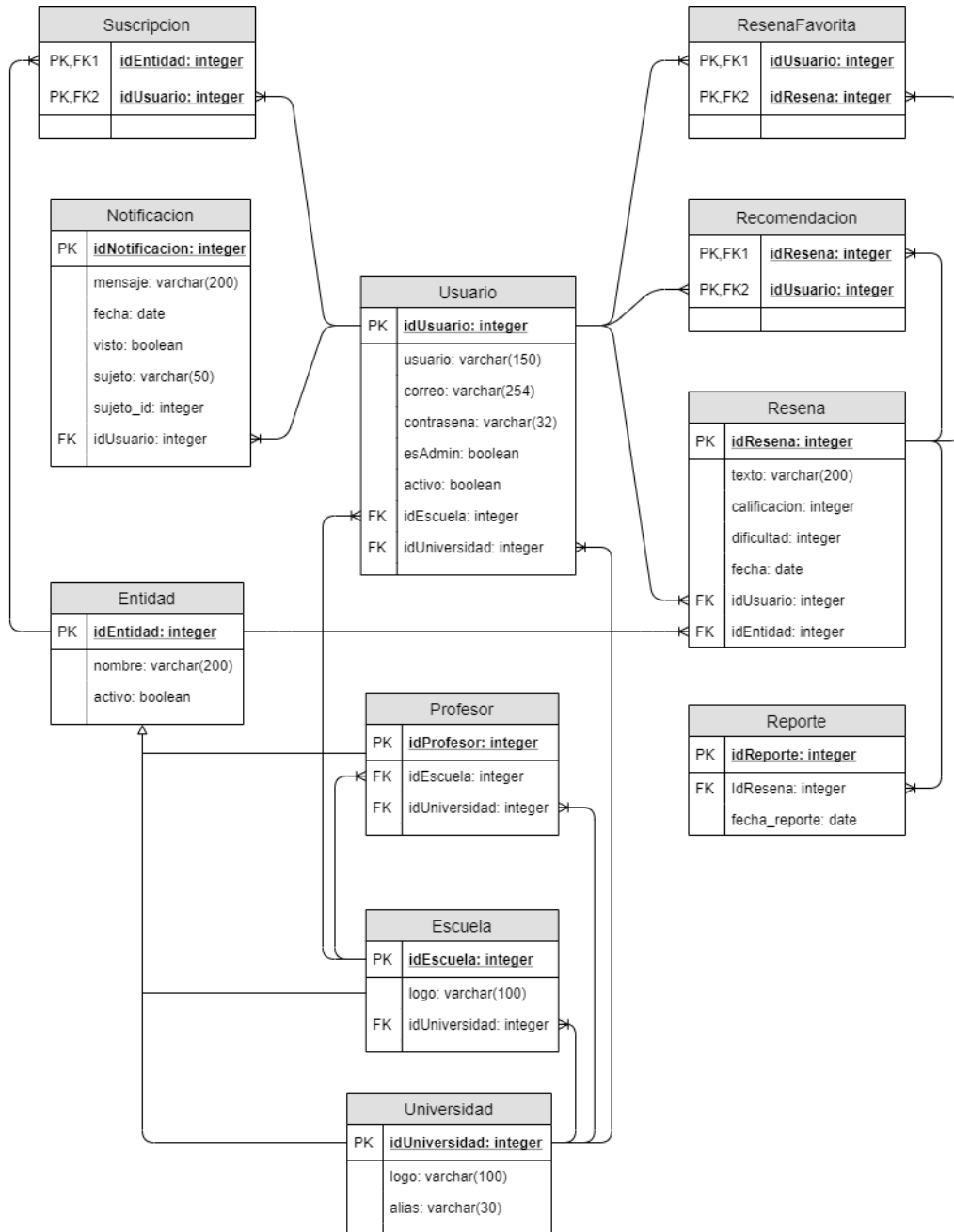


Figura 36. Diagrama de actividades para búsqueda de entidades



**6.2.4. Modelo de base de datos.** A continuación se muestra el modelo de base de datos de la aplicación, junto con una posterior descripción breve de cada tabla ahí presente.

**Figura 37.** Modelo de base de datos



**6.3. Arquitectura del software**

Para una mejor separación de las diferentes partes lógicas del software, resulta útil la elección del patrón Modelo-Vista-Controlador que realiza una buena asignación de responsabilidades de distintas partes del sistema.

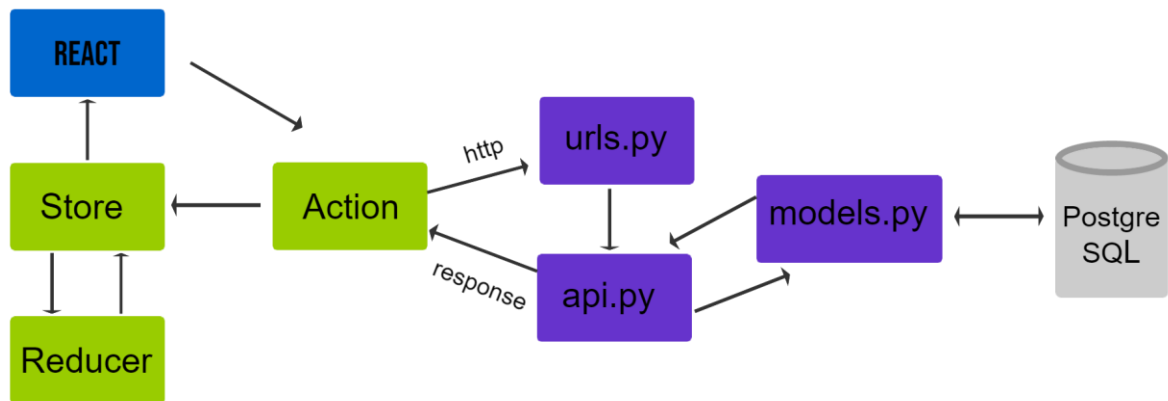
**6.3.1. Modelo-vista-controlador (MVC).** Al momento de crear una aplicación web pueden verse involucrados varios lenguajes de programación, este patrón facilita la realización de cambios al software de manera que no se vea afectado en gran medida otros sectores del código completo. Como la librería ReactJS y el framework Django fueron creados con este patrón en mente su integración con la arquitectura ya fue implementada.

**6.3.2. Modelo.** Dentro del software el modelo estaría comprendido por lo que es la base de datos Postgres, ya que es aquí donde se almacenan todos los datos ingresados desde la interfaz gráfica, y se realiza su creación, selección, modificación y eliminación.

**6.3.3. Vista.** La vista estaría comprendida por la aplicación creada con la librería de ReactJS, es decir, todo lo que se define como la interfaz gráfica de usuario. Aquí se escuchan los eventos realizados por el usuario que mandan una petición a la API creada con Django.

**6.3.4. Controlador.** El controlador está formado por la API construida con ayuda del framework Django y la librería de Python Django REST Framework. Se escuchan peticiones HTTP y se devuelven respuestas HTTP después de alguna consulta a la base de datos (el modelo).

Figura 38. Arquitectura del software



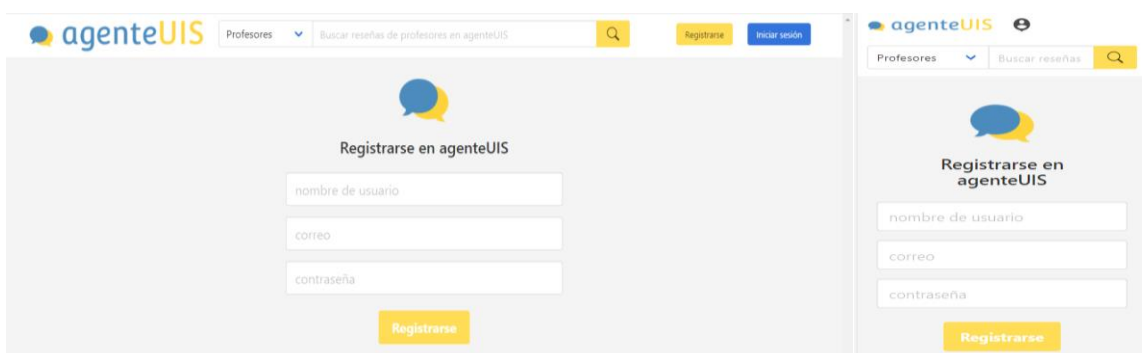
#### 6.4. Implementación de los prototipos

A continuación se describen dos prototipos, sus vistas finales y las funciones que ejecutan.

**6.4.1. Primer prototipo.** Para el primer prototipo se implementó la API de Django y la creación de la gran mayoría de los requerimientos funcionales listados, a su vez se muestra el responsive propuesto entre los requerimientos no funcionales:

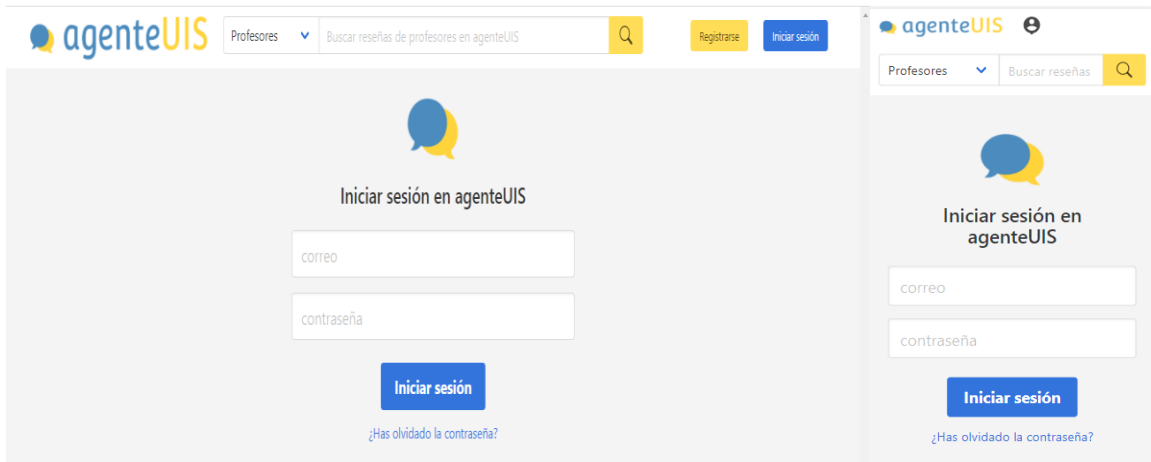
- Registro de usuario

Figura 39. Vista de escritorio y móvil de registro de usuario



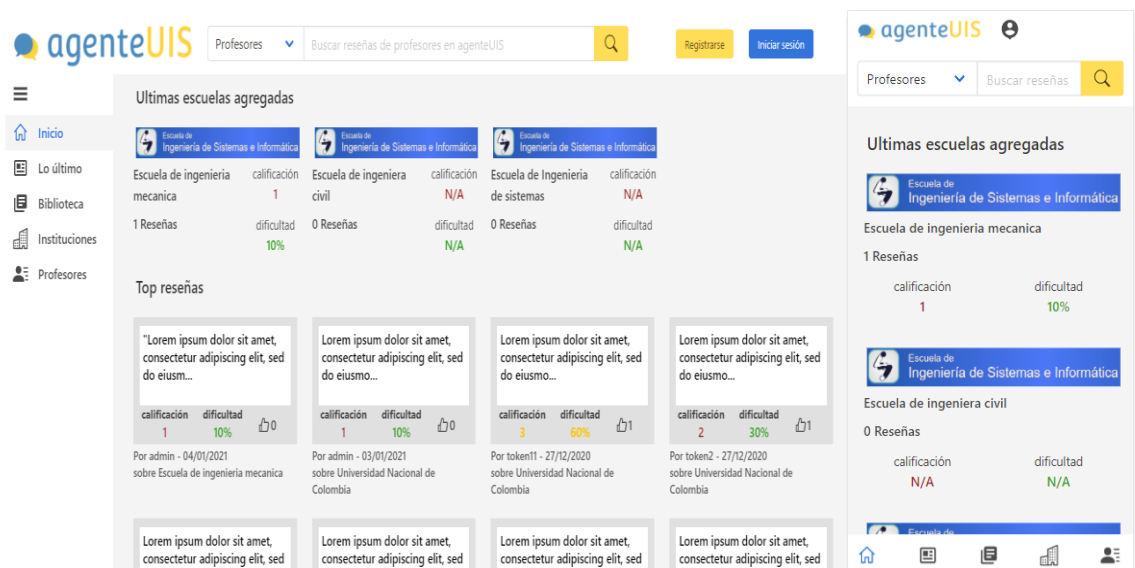
- Inicio de sesión

**Figura 40.** Vista de escritorio y móvil de inicio de sesión



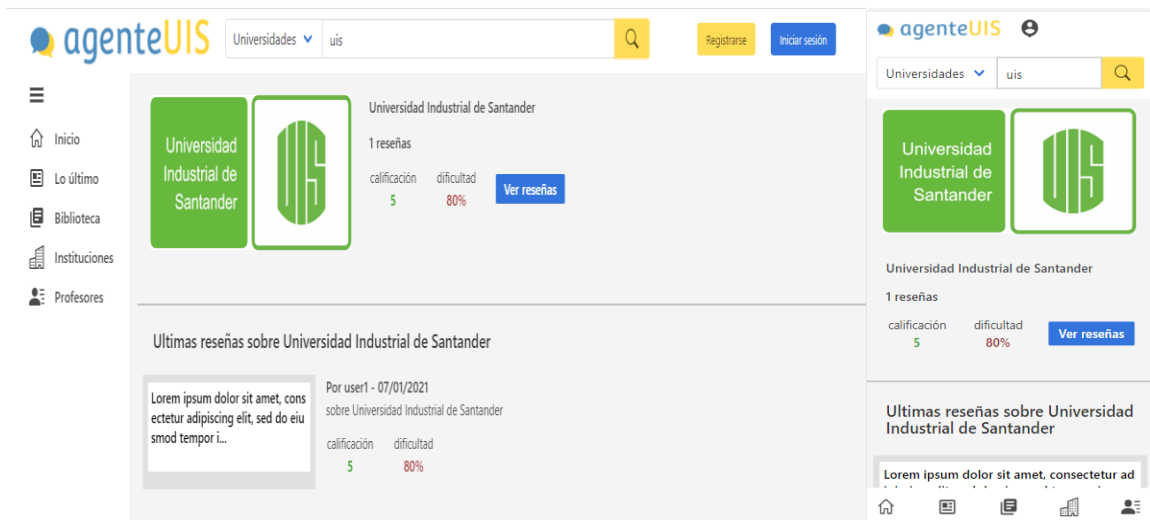
- Visualizar últimas escuelas agregadas y mejores reseñas

**Figura 41.** Vista de escritorio y móvil de página principal



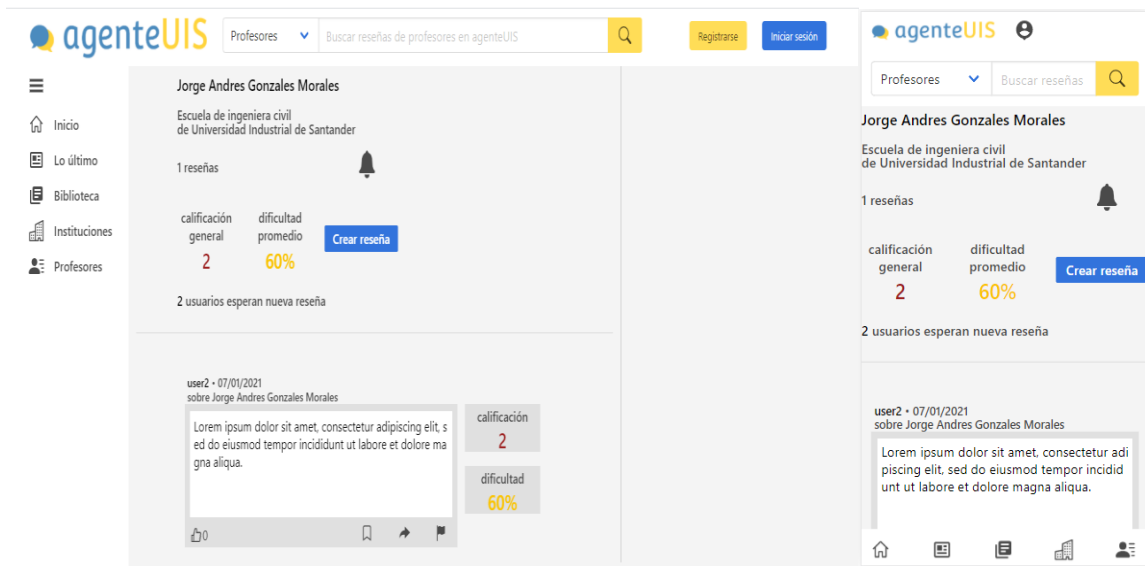
- Filtrar búsqueda de entidades

**Figura 42.** Vista de escritorio y móvil de resultados de búsqueda



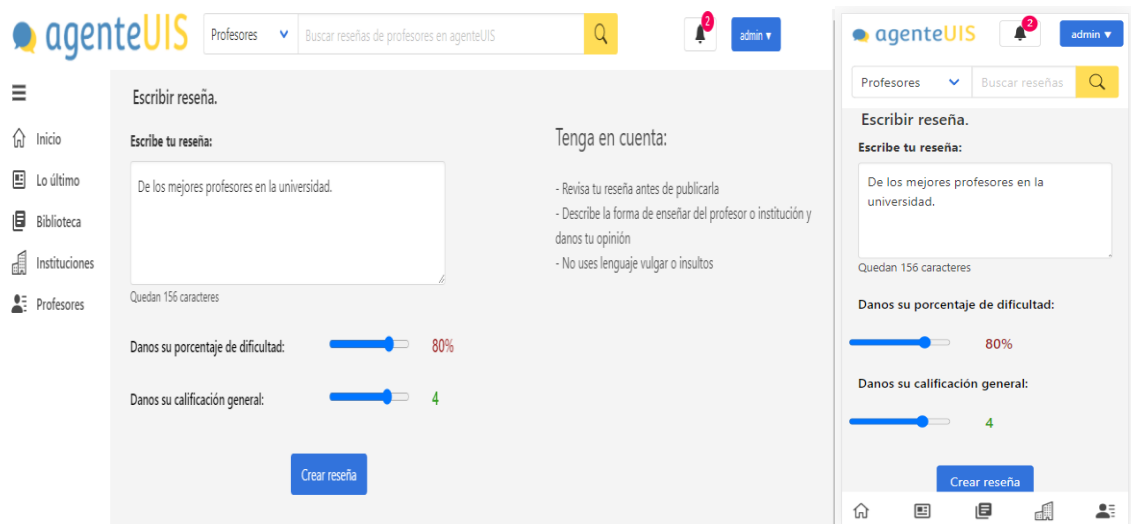
- Visualizar información de entidades y sus reseñas

**Figura 43.** Vista de escritorio y móvil de perfiles de entidades



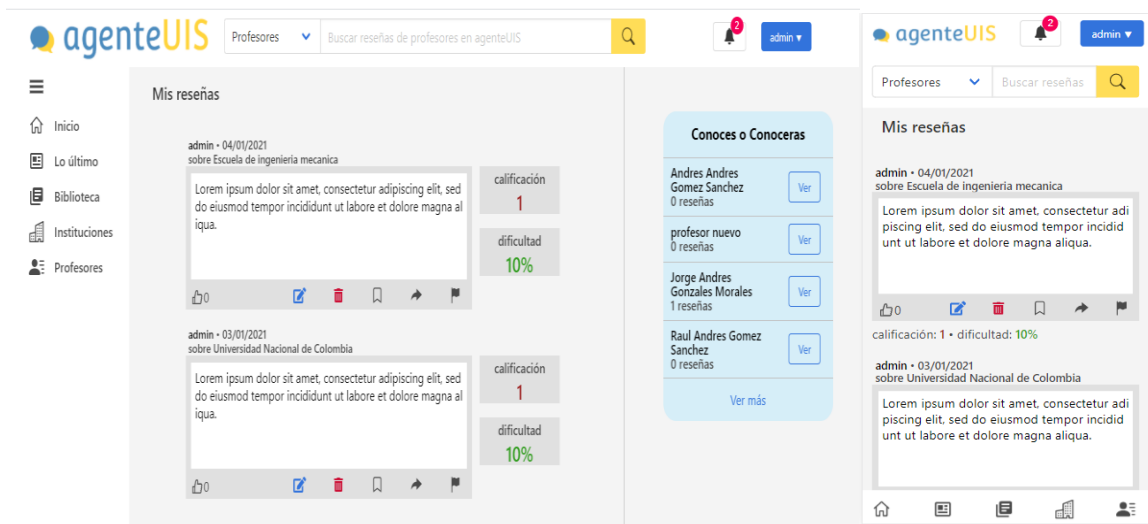
- Crear reseña

**Figura 44.** Vista de escritorio y móvil de creación de reseña

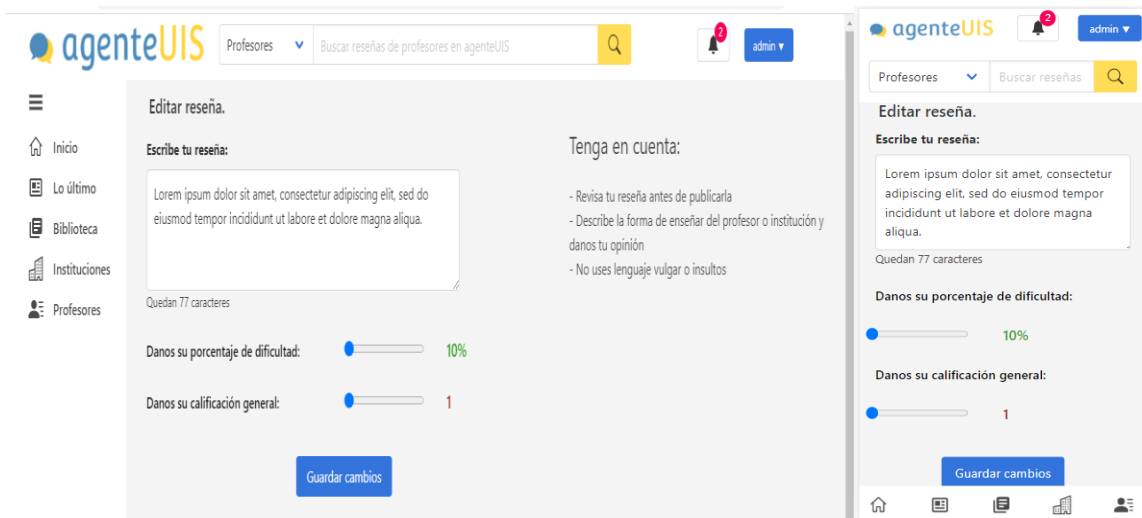


- Gestión de reseñas

**Figura 45.** Vista de escritorio y móvil de reseñas de usuario

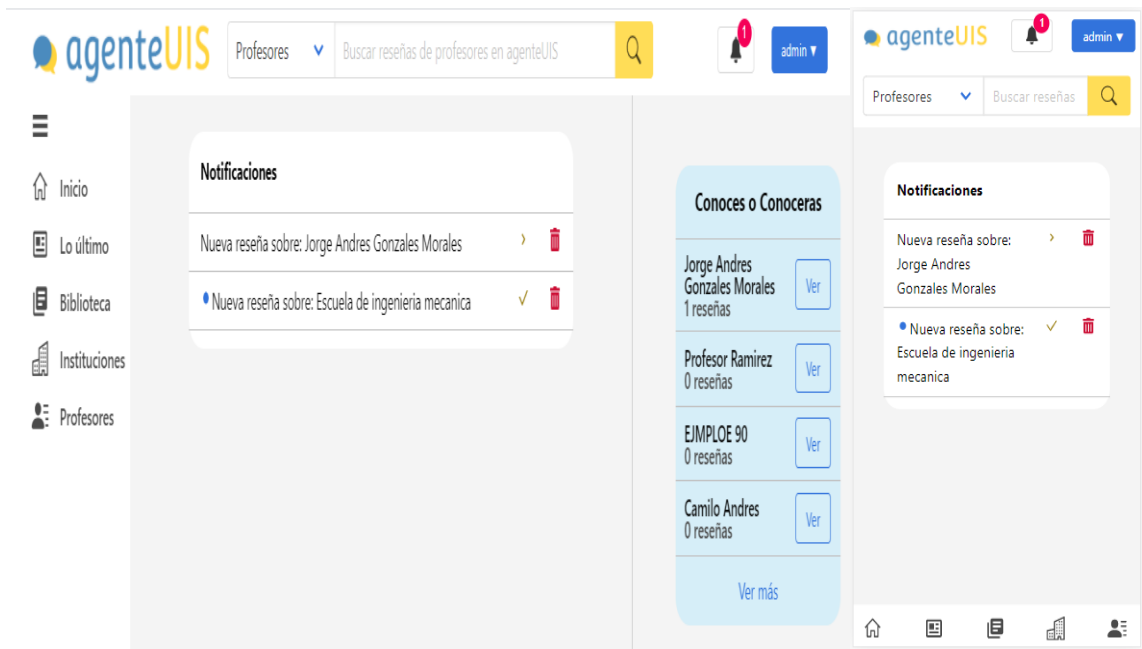


**Figura 46.** Vista de escritorio y móvil de edición de reseñas



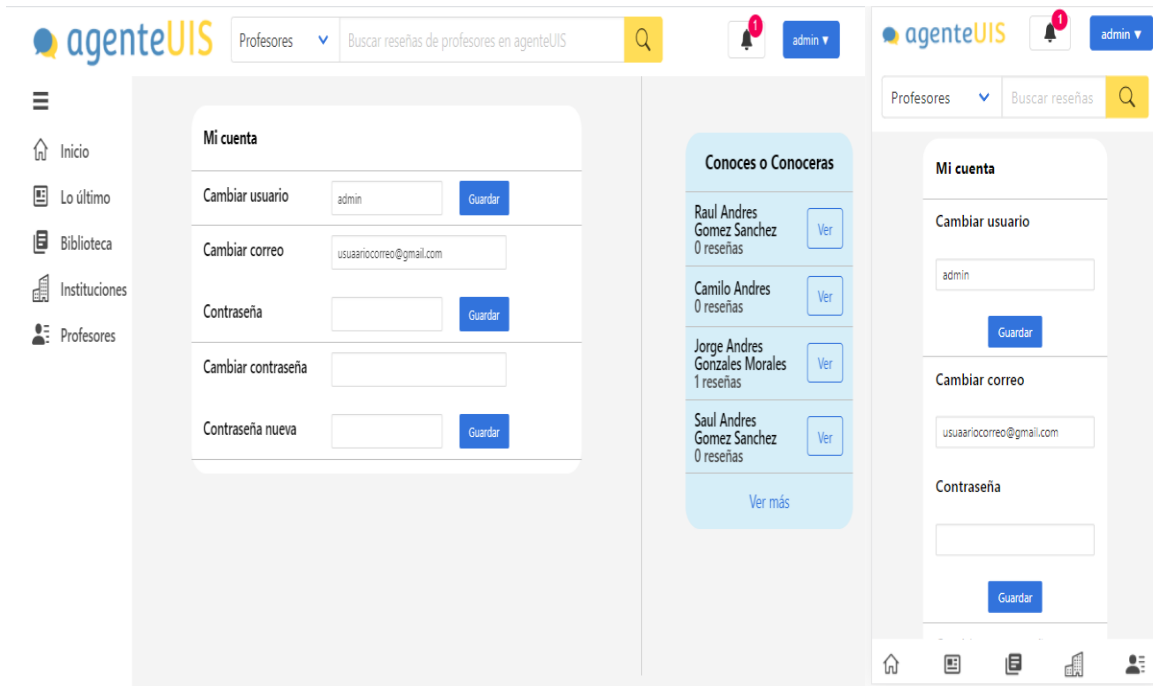
- Recibir notificaciones de nueva reseña

**Figura 47.** Vista de escritorio y móvil de notificaciones de nueva reseña



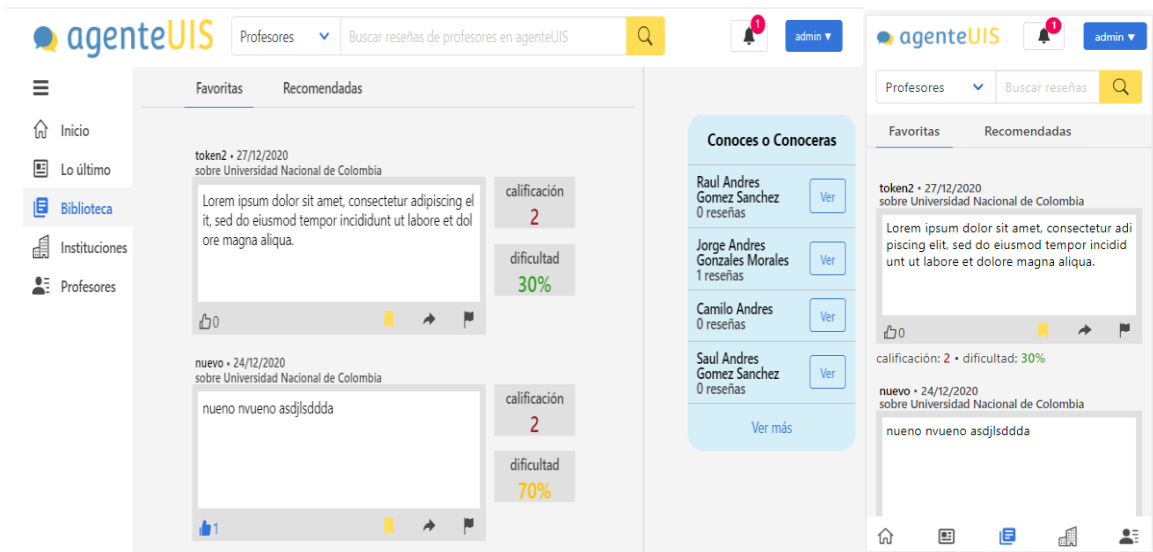
- Editar cuenta de usuario

Figura 48. Vista de escritorio y móvil de edición de cuenta



- Ver reseñas recomendadas y guardadas

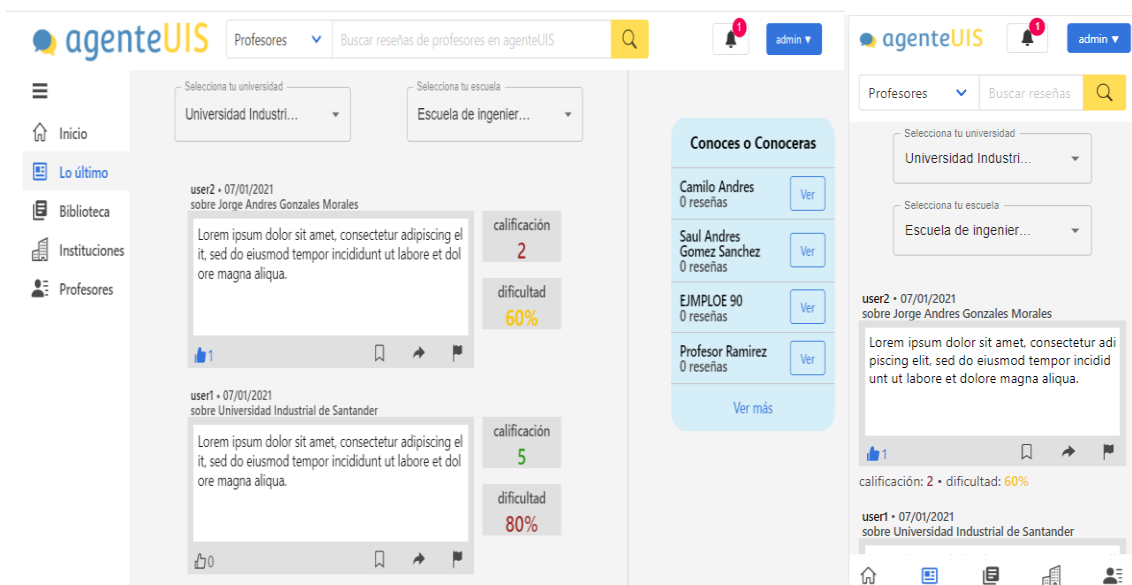
Figura 49. Vista de escritorio y móvil de biblioteca



**6.4.2. Segundo prototipo.** En esta fase se agrega la funcionalidad para mostrar el feed de usuario después de haber seleccionado escuela y universidad, junto con la creación de reportes, manejo del error 404 y la visualización de “esqueletos” mientras cargan elementos principales.

- Feed de usuario

**Figura 50.** Vista de escritorio y móvil de feed de usuario



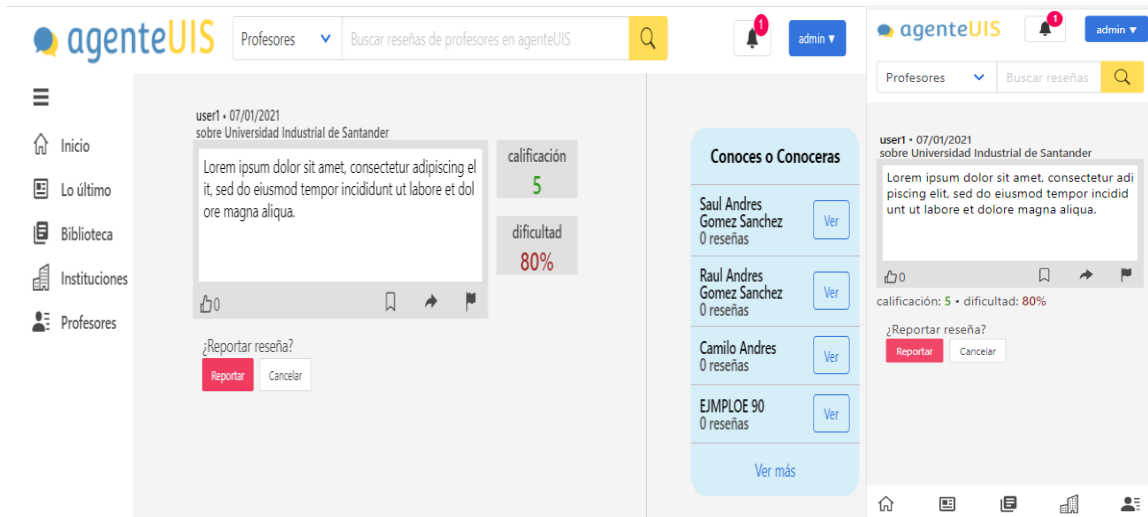
- Página 404

**Figura 51.** Vista de escritorio y móvil de página 404



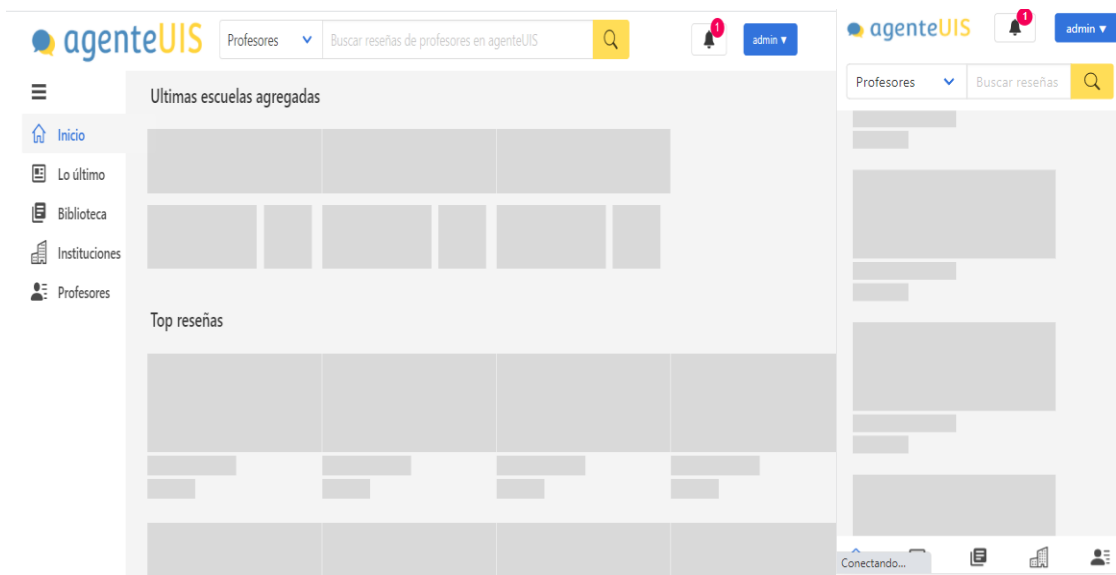
- Reportar reseña

**Figura 52.** Vista de escritorio y móvil para reportar reseña



- Esqueletos de carga

**Figura 53.** Vista de escritorio y móvil de esqueletos de carga





**Tabla 31. Prueba registrar usuario**

Registrar usuario							
<b>Correo</b>	-	mail@gm ail.com	mail@gm ail.com	badmail	mail@gm ail.com	mail@gm ail.com	mail@gm ail.com
<b>Usuario</b>	User	-	user	user	us er	user	user
<b>Contraseña</b>	password 1	password1	-	password 1	password	pass	password1
<b>Respuesta</b>	Mensaje: “Correo: este campo es requerido ”	Mensaje: “Usuario: este campo es requerido”	Mensaje: “Contrase ña: este campo es requerido”	Mensaje: “Correo: Ingrese una dirección de correo válida”	Mensaje: “Usuario: el usuario solo puede contener caracteres y números ”	Mensaje: “Contrase ña: debe tener 8 o más caracteres, y letras y números”	Registro de usuario e inicio de sesión.
<b>Resultado</b>	Correcto	Correcto	Correcto	Correcto	Correcto	Correcto	Correcto

**Tabla 32. Prueba cerrar sesión**

Cerrar sesión	
<b>Opción cerrar</b>	Seleccionar cerrar sesión

<b>sesión</b>	
<b>Respuesta</b>	Dstrucción de token de autenticación
<b>Resultado</b>	Correcto

**Tabla 33.** Prueba cambio de nombre de usuario

Cambio de nombre de usuario			
<b>Usuario nuevo</b>	-	us er	new_user
<b>Respuesta</b>	Mensaje: “usuario: este campo es requerido”	Mensaje: “usuario: el usuario solo puede contener letras y números”	Cambio de nombre de usuario
<b>Resultado</b>	Correcto	Correcto	Correcto

**Tabla 34.** Prueba cambio de correo

Cambio de correo			
<b>Correo nuevo</b>	-	mail@gmail.co m	new_mail@g mail.com
<b>Contraseña</b>	password	-	password

<b>actual</b>			
<b>Respuesta</b>	Mensaje: “correo: este campo no puede estar en blanco”	Mensaje: “contraseña: este campo no puede estar en blanco”	Cambio de correo
<b>Resultado</b>	Correcto	Correcto	Correcto

**Tabla 35.** Prueba recuperación de contraseña

Recuperación de contraseña			
<b>Correo</b>	bad_mail@ga mil.com	-	mail@gmail.c om
<b>Respuesta</b>	Mensaje: “correo: no existe un usuario con el mail dado.”	Mensaje: “correo: este campo no puede estar en blanco”	Envío de link de cambio de contraseña
<b>Resultado</b>	Correcto	Correcto	Correcto

**Tabla 36.** Prueba agregar universidades

Agregar universidades desde cuenta administrativa				
<b>Nombre</b>	-	Universidad Industrial de Santander	Universidad Industrial de Santander	Universidad Industrial de Santander
<b>Logo</b>	Imagen	-	Imagen	Imagen
<b>Alias</b>	UIS	UIS	-	UIS
<b>Respuesta</b>	Mensaje: “Nombre: este campo es requerido”	Mensaje: “Logo: este campo es requerido”	Mensaje: “Alias: este campo es requerido”	Universidad agregada
<b>Resultado</b>	Correcto	Correcto	Correcto	Correcto

**Tabla 37.** Prueba agregar escuelas

Agregar escuela desde cuenta administrativa				
<b>Nombre</b>	-	Escuela de ingeniería de sistemas	Escuela de ingeniería de sistemas	Escuela de ingeniería de sistemas
<b>Logo</b>	Imagen	-	Imagen	Imagen
<b>Universidad</b>	Universidad	Universidad	-	Universidad

	Industrial de Santander	Industrial de Santander		Industrial de Santander
<b>Respuesta</b>	Mensaje: “Nombre: este campo es requerido”	Mensaje: “Logo: este campo es requerido”	Mensaje: “Universidad : este campo es requerido”	Escuela agregada
<b>Resultado</b>	Correcto	Correcto	Correcto	Correcto

**Tabla 38.** Prueba agregar profesor

Agregar profesor desde cuenta administrativa				
<b>Nombre</b>	-	Jorge Ramírez	Jorge Ramírez	Jorge Ramírez
<b>Escuela</b>	Escuela de ingeniería de sistemas	-	Escuela de ingeniería de sistemas	Escuela de ingeniería de sistemas
<b>Universidad</b>	Universidad Industrial de Santander	Universidad Industrial de Santander	-	Universidad Industrial de Santander
<b>Respuesta</b>	Mensaje: “Nombre: este campo es	Mensaje: “Escuela: este campo es	Mensaje: “Universidad : este campo	Profesor agregado

	requerido”	requerido”	es requerido”	
<b>Resultado</b>	Correcto	Correcto	Correcto	Correcto

**Tabla 39.** Prueba eliminar usuario

Eliminar usuario	
<b>Botón eliminar usuario</b>	Seleccionar eliminar
<b>Respuesta</b>	Eliminación del usuario de la base de datos
<b>Resultado</b>	Correcto

**Tabla 40.** Prueba agregar reseña

Agregar reseña				
<b>Texto</b>	-	“Texto con 20 caracteres”	“Texto con más de 200 caracteres”	Buen profesor a lo largo de todo el semestre
<b>Calificación</b>	4	4	4	4
<b>Dificultad</b>	60	60	60	60
<b>Respuesta</b>	Mensaje:	Mensaje:	Mensaje:	Reseña creada

	“Texto: este campo es requerido”	“Texto: Debe tener al menos 20 caracteres”	“Texto: Debe ser menor a 200 caracteres”	
<b>Resultado</b>	Correcto	Correcto	Correcto	Correcto

**Tabla 41.** Prueba guardar reseña

Guardar reseña	
<b>Botón de guardar</b>	Clic en botón para guardar
<b>Respuesta</b>	Reseña guardada en favoritos
<b>Resultado</b>	Correcto

**Tabla 42.** Prueba compartir reseña

Compartir reseña	
<b>Botón para compartir</b>	Clic en botón para compartir
<b>Opciones para compartir</b>	Seleccionar opción para compartir
<b>Respuesta</b>	Reseña compartida

<b>Resultado</b>	Correcto
------------------	----------

**Tabla 43.** *Prueba recomendar reseña*

Recomendar reseña	
<b>Botón de recomendar</b>	Clic en botón para recomendar
<b>Respuesta</b>	Reseña guardada en recomendadas y mostrar nueva cantidad de recomendaciones
<b>Resultado</b>	Correcto

**Tabla 44.** *Prueba consultar reseñas recomendadas*

Consultar reseñas recomendadas	
<b>Opción de menú Biblioteca</b>	Clic en Biblioteca y en Recomendadas
<b>Respuesta</b>	Vista con las reseñas recomendadas por el usuario
<b>Resultado</b>	Correcto

**Tabla 45.** *Prueba suscribirse a entidades*

Suscribirse a Universidades, Escuelas y profesores	
<b>Campana para suscribirse</b>	Clic en campana
<b>Respuesta</b>	Usuario suscrito a entidad visualizada
<b>Resultado</b>	Correcto

**Tabla 46.** *Prueba ver notificaciones*

Ver notificaciones de nuevas reseñas	
<b>Campana de notificaciones</b>	Clic en campana
<b>Respuesta</b>	Vista con lista de todas las notificaciones para el usuario
<b>Resultado</b>	Correcto

**Tabla 47.** *Prueba búsqueda de entidades*

Filtrar búsqueda de universidades, escuelas y profesores			
<b>Categoría</b>	Universidades	Escuelas	Profesores

<b>Búsqueda</b>	Universidad Industrial de Santander	Escuela de Ingeniería de sistemas	Jorge Ramírez
<b>Respuesta</b>	Resultados de búsqueda en universidades o No se encontraron resultados para su búsqueda.	Resultados de búsqueda en escuelas o No se encontraron resultados para su búsqueda.	Resultados de búsqueda en profesores o No se encontraron resultados para su búsqueda.
<b>Resultado</b>	Correcto	Correcto	Correcto

**Tabla 48.** Prueba visualizar reseñas de entidades

Visualizar reseñas de universidades, escuelas y profesores	
<b>Listado de entidades</b>	Seleccionar entidad
<b>Respuesta</b>	Visualizar listado de reseñas de la entidad seleccionada o No hay reseñas aún.
<b>Resultado</b>	Correcto

**Tabla 49.** Prueba visualizar reseñas más recomendadas en inicio

Visualizar reseñas más recomendadas en página principal	
<b>Opción de menú</b>	Seleccionar Inicio

<b>Inicio</b>	
<b>Respuesta</b>	Visualizar reseñas según recomendaciones recibidas
<b>Resultado</b>	Correcto

**Tabla 50.** Prueba visualizar últimas escuelas agregadas

Visualizar últimas escuelas agregadas en página principal	
<b>Opción de menú</b>	Seleccionar Inicio
<b>Inicio</b>	
<b>Respuesta</b>	Visualizar últimas escuelas agregadas
<b>Resultado</b>	Correcto

**Tabla 51.** Prueba ver profesores probables

Ver profesores probables			
<b>Sesión</b>	Cerrada	Iniciada	Iniciada
<b>Universidad y escuela de usuario</b>	-	-	Universidad Industrial de Santander y Escuela de ingeniería de sistemas

<b>Respuesta</b>	No se muestra vista de profesores probables	Elige tu universidad y escuela para ver profesores probables	Vista de profesores probables
<b>Resultado</b>	Correcto	Correcto	Correcto

6.5.2. Pruebas del segundo prototipo

Tabla 52. Prueba reportar reseña

Reportar reseña	
<b>Botón de reportar</b>	Clic en reportar
<b>Respuesta</b>	Reporte creado en base de datos
<b>Resultado</b>	Correcto

Tabla 53. Prueba visualizar reportes

Visualizar reportes en panel de administrador	
<b>Listado de opciones de gestión para administrador</b>	Clic en reportes
<b>Respuesta</b>	Visualizar listado de reportes

<b>Resultado</b>	Correcto
------------------	----------

**Tabla 54.** Prueba feed de usuario

Feed de usuario		
<b>Opción de menú</b> <b>Lo último</b>	Seleccionar Lo último	Seleccionar Lo último
<b>Universidad y escuela de usuario</b>	-	Universidad Industrial de Santander y Escuela de Ingeniería de sistemas
<b>Respuesta</b>	Elige para ver las reseñas de tu universidad, tu escuela y los profesores de tu escuela	Visualizar reseñas según su ranking
<b>Resultado</b>	Correcto	Correcto

### 6.6. Configuraciones en Amazon web services

En esta parte se describe el proceso de despliegue de una aplicación de ReactJS y Django en diferentes servicios de Amazon Web Services.

**6.6.1. Aplicación de React en S3.** Para poder usar el servicio S3 para alojar archivos estáticos de una aplicación web se necesitan los siguientes prerequisites:

- Cuenta en Amazon Web Services.
- Usuario creado en IAM para administrar el servicio S3.
- AWS CLI instalado en local.
- Credenciales de Amazon Web Services en local.

1. Con una cuenta ya creada, se puede proseguir a iniciar sesión en la consola principal de AWS.

2. En servicios se busca IAM.

3. En la consola de IAM se necesita crear un usuario para manejar el bucket S3.

Se da clic en agregar usuario.

4. Al usuario se le asigna un nombre y se le selecciona acceso programático.

5. Después, en permisos, se le asigna la política ya existente de AdministratorAccess al usuario.

6. Se prosigue a crear el usuario y guardar su llave de acceso secreta e identificación de la llave.

7. Teniendo instalado el AWS CLI localmente, se puede proseguir a configurarlo con las credenciales descargadas usando el comando **aws configure** en una terminal.

8. Se da la llave de acceso y su identificación apenas es solicitada en la terminal.

9. De nuevo en la consola de AWS se busca el servicio S3.

10. Ya en la consola del servicio S3 se selecciona crear bucket. Al bucket se le da un nombre único y en permisos se selecciona **Grant public read access to this bucket** y finalmente crear bucket.

11. Dentro del bucket creado se seleccionan sus propiedades y se escoge la opción para hosting de página web estática.

12. Después en permisos se selecciona editar la política del bucket y se pone lo siguiente, para conceder acceso de lectura al bucket:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPublicReadAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::example-bucket/*"
      ]
    }
  ]
}
```

13. Ya para finalizar, se puede crear un comando interno para el proyecto de React dentro del archivo package.json:

```
“deploy”: “aws s3 sync build/ s3://example-bucket --acl public-read”
```

Con esto se puede hacer **npm run deploy**, y se realizará el despliegue de la aplicación de React en el bucket S3 (Linton, 2018).

**6.6.2. Aplicación de Django en Elastic Beanstalk.** Antes de poder desplegar la aplicación de Django en Elastic Beanstalk se necesitan los siguientes prerequisites:

- Tener una aplicación de Elastic BeanStalk creada
- Crear un ambiente usado por la aplicación con Python 3.7 y Amazon Linux 2

1. En la raíz del proyecto de Django se crea la carpeta `.ebextensions` y dentro se crea el archivo `django.conf` con lo siguiente:

```
option_settings:
```

```
aws:elbv2:listener:80:
```

```
    DefaultProcess: http
```

```
    ListenerEnabled: 'true'
```

```
    Protocol: HTTP
```

```
    Rules: ws
```

```
aws:elbv2:listenerrule:ws:
```

```
    PathPatterns: /ws/*
```

```
    Process: websocket
```

```
    Priority: 1
```

```
aws:elasticbeanstalk:environment:process:http:
```

```
    Port: '80'
```

```
    Protocol: HTTP
```

```
aws:elasticbeanstalk:environment:process:websocket:
```

```
    Port: '5000'
```

```
    Protocol: HTTP
```

Con esto se le dice a Elastic BeanStalk que escuche peticiones en el puerto 80 y si llega una petición websocket `/ws/*` se pasa la petición al puerto 5000.

2. Finalmente, otra vez en la raíz del proyecto, dentro de un archivo creado llamado Procfile se pone lo siguiente:

```
web:    gunicorn    --bind    :8000    --workers    3    --threads    2
<django_app>.wsgi:application
websocket: daphne -b 0.0.0.0 -p 5000 <django_app>.asgi:application
```

Con esto se crean los servidores que se harán cargo de las peticiones HTTP y de WebSocket, respectivamente (Capacillo, 2020).

## 7. Conclusiones

Se logró diseñar una aplicación web de reseñas estudiantiles teniendo en cuenta todos los requerimientos listados en los objetivos planteados. Y aunque el proyecto trate del desarrollo de un software, no se dejó de lado su justificación principal, plasmando y dejando en claro la importancia que tienen los estudiantes a la hora de dar su opinión sobre los profesores que ven en su vida universitaria. Se trata de un derecho que siempre deberían tener y poder expresar públicamente.

Se comprobó que el estilo de arquitectura REST sirve muy bien para separar largos proyectos en dos, una parte para el manejo de la interfaz gráfica de usuario y otra encargada del manejo de los datos, esto claramente ayuda a realizar grandes cambios al sistema sin afectar por completo si se llega a producir un error. También se evidencia la utilidad del modelo-vista-controlador al delegar muy bien diferentes tareas complejas entre sus diferentes partes, esto ayudará a un posible futuro escalamiento que se quiera realizar.

Por último es importante mencionar que los frameworks como Django y React han demostrado ser poderosas herramientas que han llegado para quedarse; facilitando el trabajo de millones de desarrolladores alrededor del mundo en su tarea de crear aplicaciones complejas

con mayor rapidez y facilidad que años pasados, cuando esta clase de herramientas no estaban tan presentes. Hay que darle crédito por su crecimiento exponencial de los últimos años a la enorme comunidad de desarrolladores de software de código abierto que sigue dando soporte y evolucionando estos frameworks usados en múltiples proyectos.

## **8. Recomendaciones**

Para un continuo mejoramiento del software y poder diseñar nuevos prototipos se dejan las siguientes recomendaciones:

- Mejorar el sistema de reportes de manera que se pueda notificar al usuario cuya reseña fue reportada.
- Implementar la búsqueda de entidades entre todas las clases de manera que se pueda buscar entre: todas, profesores, universidades y escuelas.
- Diseñar una aplicación móvil que conecte con la misma API para alcanzar a más usuarios.

### Referencias Bibliográficas

- Abramov, D. (2015). Quick Start. <https://react-redux.js.org/introduction/quick-start>
- Alvarez, M. (2018). Qué es Redux. <https://desarrolloweb.com/articulos/que-es-redux.html>
- Alvarez, M. (2019). Qué es React. Por qué usar React. <https://desarrolloweb.com/articulos/que-es-react-motivos-uso.html>
- Amazon Web Services Inc. (s.f.). Amazon EC2. <https://aws.amazon.com/es/ec2>
- Amazon Web Services Inc. (s.f.). Amazon Relational Database. <https://aws.amazon.com/es/rds>
- Amazon Web Services Inc. (s.f.). Amazon S3. <https://aws.amazon.com/es/s3>
- Amazon Web Services Inc. (s.f.). AWS Elastic Beanstalk. <https://aws.amazon.com/es/elasticbeanstalk>
- Amazon Web Services Inc. (s.f.). Informática en la nube con AWS. <https://aws.amazon.com/es/what-is-aws>
- Azaustre, C. (2015). ¿Qué es la autenticación basada en Token? <https://carlosazaustre.es/que-es-la-autenticacion-con-token>
- Bulma. (s.f.). Bulma. <https://bulma.io>
- Capacillo, J. (2020). How to deploy Django Channels 2.x on AWS Elasticbeanstalk using Amazon Linux 2 AMI. <https://blog.xoxzo.com/2020/07/17/deploy-django-channels-in-elasticbeanstalk>
- Carrillo de Leon, A. (2018). Modelos de desarrollo de software. <https://ivan395.github.io/Web/modelos.html>
- Crawford, S. (s.f.). How RateMyProfessors Works. <https://computer.howstuffworks.com/internet/social-networking/networks/ratemyprofessors.htm>

- Dabit, N. (2016). Beginner's Guide to React Router.  
<https://www.freecodecamp.org/news/beginner-s-guide-to-react-router-53094349669>
- Django Software Foundation. (2005). Meet Django. <https://www.djangoproject.com>
- Django Software Foundation. (2018). Django Channels.  
<https://channels.readthedocs.io/en/latest>
- Django-rest-framework.org. (s.f.). Djoser. <https://www.django-rest-framework.org/api-guide/authentication/#djoser>
- Facebook Inc. (s.f.). Crear una nueva aplicación React.  
<https://es.reactjs.org/docs/create-a-new-react-app.html>
- Facebook Inc. (s.f.). React. <https://es.reactjs.org>
- García, J. (2017). Single-Page Application, todo un website desde única página.  
<https://www.arsys.es/blog/programacion/spa-unica-pagina>
- Google. (s.f.). ¿Qué es un feed?  
<https://support.google.com/adsense/answer/9189559?hl=es>
- Google Play. (s.f.). DeTodoUIS.  
<https://play.google.com/store/apps/details?id=com.rafaicf.detodouisapp>
- Gori, R. (2020). Working with Redis in Python with Django.  
<https://stackabuse.com/working-with-redis-in-python-with-django>
- Igi-global.com. (s.f.). What is Student Ratings. <https://www.igi-global.com/dictionary/student-ratings/28575>
- Linton, J. (2018). Deploy ReactJS App with S3 Static Hosting.  
<https://medium.com/serverlessguru/deploy-reactjs-app-with-s3-static-hosting-f640cb49d7e6>

- Lucas, J. (2019). Qué es NodeJS y para qué sirve. <https://openwebinars.net/blog/que-es-nodejs>
- Mariano, G., Hammonds, F., Chambers, S., Ammons, G., y Sippel, K.(2018). Using Student Feedback to Help Improve Teaching in the College Classroom. <https://www.igi-global.com/chapter/using-student-feedback-to-help-improve-teaching-in-the-college-classroom/199437>
- Mekhiya, K. (2016). Software PROTOTYPING — Worth or Not?. <https://medium.com/content-writing/software-prototyping-worth-or-not-369a19ba6758>
- Microsoft. (s.f.) Getting Started. <https://code.visualstudio.com/docs>
- Mozilla. (2020). JavaScript. <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Muradas, Y. (2020). Qué es ECMAScript. <https://openwebinars.net/blog/que-es-ecmascript>
- Ortiz, A. (2018). Qué es un framework en informática o programación. <https://blog.hostdime.com.co/que-es-un-framework-informatica-programacion>
- PostgreSQL Global Development Group. (s.f.). What is PostgreSQL? <https://www.postgresql.org/about>
- Python Software Foundation. (2001). What is Python? <https://docs.python.org/3/faq/general.html#what-is-python>
- Raji, S. (2017). What is the Model View Controller design pattern? <https://medium.com/@ormrepo/what-is-the-model-view-controller-design-pattern-e951816217d9>
- Simoos, C. (2019). REST vs WebSocket. ¿Hay diferencias? <https://www.chiyanasimoos.com/blog/rest-vs-websocket-hay-diferencias>

Statista. (2020). Most used programming languages among developers worldwide.

<https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages>

Theall, M. (2003). Student Ratings: Myths vs. Research Evidence.

<https://cft.vanderbilt.edu/articles-and-essays/the-teaching-forum/student-ratings-myths-vs-research-evidence>

Wikipedia. (s.f.). RateMyTeachers. <https://en.wikipedia.org/wiki/RateMyTeachers>

Wikipedia. (s.f.). RateMyProfessors.

<https://en.wikipedia.org/wiki/RateMyProfessors.com>