

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias.
Ingeniería de Sistemas y Computación.
Proyecto de Grado.

Análisis comparativo de dos procesos de definición de requisitos para el desarrollo de un sistema web

Santiago Coca Rojas

Director: Juan Carlos Martínez Arias

15 de marzo de 2021



Santiago de Cali, 15 de marzo de 2021.

Señores

Pontificia Universidad Javeriana Cali.

Dr. Gerardo Mauricio Sarria

Director Carrera de Ingeniería de Sistemas y Computación.

Cali.

Cordial Saludo:

Por medio de la presente me permito informarle que he revisado el proyecto de grado “Análisis comparativo de dos procesos de definición de requisitos para el desarrollo de un sistema web” del estudiante Santiago Coca Rojas (cod: 8923527), del cual soy Director y lo considero finalizado y listo para su sustentación.

Atentamente,

Juan Carlos Martínez Arias
Director de Trabajo de Grado
Pontificia Universidad Javeriana Cali

Santiago de Cali, 15 de marzo de 2021.

Señores

Pontificia Universidad Javeriana Cali.

Dr. Gerardo Mauricio Sarria

Director Carrera de Ingeniería de Sistemas y Computación.

Cali.

Cordial Saludo.

Me permito presentar a su consideración el proyecto de grado titulado “Análisis comparativo de dos procesos de definición de requisitos para el desarrollo de un sistema web” con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el proyecto de grado y posteriormente optar al título de Ingeniero de Sistemas y Computación.

Atentamente,

Santiago Coca Rojas

Código: 8923527

Resumen

El proceso de definición de requisitos de un sistema, también conocido como ingeniería de requisitos, generalmente es la base de todo proceso de desarrollo de software, ya que sienta las bases del resultado que se quiere obtener de acuerdo a lo que quiere el cliente y las partes interesadas. En este proyecto, se realizó un análisis detallado del proceso de definición de requisitos utilizando dos metodologías de desarrollo diferentes: metodologías tradicionales (proceso realizado por los estudiantes del curso de Ingeniería de Requisitos, perteneciente al programa de maestría en Ingeniería de Software de la Pontificia Universidad Javeriana Cali) y metodologías basadas en la experiencia de usuario (metodologías UX), lo cual llevó a la realización de un análisis comparativo de ambos procesos y así se pudo determinar cuál de los dos procesos de definición de requisitos es el más adecuado a la hora de enfrentarse a un desarrollo de una aplicación web.

Palabras Clave: Requisitos, definición de requisitos, metodologías de desarrollo, experiencia de usuario, metodología tradicional, desarrollo web, ingeniería de requisitos.

Índice general

1. Análisis previo y descripción del problema	13
1.1. Planteamiento del Problema	13
1.1.1. Formulación	14
1.1.2. Sistematización	14
1.2. Objetivos	15
1.2.1. Objetivo General	15
1.2.2. Objetivos Específicos	15
1.3. Justificación	15
1.4. Marco de Referencia	16
1.4.1. Áreas Temáticas	16
1.4.2. Marco Teórico	16
1.4.3. Trabajos Relacionados	19
2. Desarrollo del Proyecto	21
2.1. Selección de la metodología de desarrollo	21
2.1.1. Condiciones previas	21
2.1.2. Criterios de evaluación	22
3. Metodología seleccionada: Agile UX	29
3.1. Descripción general	29
3.2. Proceso detallado aplicado al proyecto	29
3.2.1. Reuniones con stakeholders	29
3.2.2. Historias de usuario	30
3.2.3. Backlog funcionalidades	30
3.2.4. Backlog interacciones	31
3.2.5. Backlog unificado	31
3.2.6. Sprint backlog	31
3.2.7. Sprint	32
3.2.8. Reuniones para feedback	32
3.2.9. Pruebas funcionales y de interacción	32
3.3. Diagrama de la metodología	33
4. Elicitación de Funcionalidades e Interacciones	35
4.1. Tipos de usuarios	35
4.1.1. Aspirantes	35
4.1.2. Estudiantes	35
4.1.3. Docentes	35

4.1.4. Personal administrativo y colaboradores	35
4.2. Funcionalidades	36
4.2.1. Aspirantes	38
4.2.2. Estudiantes	40
4.2.3. Profesores	41
4.2.4. Personal Administrativo	41
4.3. Interacciones	43
4.3.1. Aspirantes	44
4.3.2. Estudiantes	45
4.3.3. Profesores	45
4.3.4. Personal adminitrativo	46
4.4. Funcionalidades e interacciones adicionales	46
4.4.1. Funcionalidades	46
4.4.2. Interacciones	47
4.4.3. Validación de funcionalidades e interacciones adicionales	47
4.5. Segundo prototipo	47
4.5.1. Entrevistas	49
4.5.2. Más funcionalidades	51
4.5.3. Más interacciones	51
5. Herramientas utilizadas para el prototipado	53
5.1. Primer prototipo	53
5.2. Segundo prototipo	54
5.2.1. React.js	54
5.2.2. Node.js	54
5.2.3. MongoDB	55
6. Análisis comparativo de dos procesos de ingeniería de requisitos	57
6.1. Proceso realizado con ingeniería de requisitos de metodologías tradicionales	57
6.1.1. Requisitos	58
6.1.2. Prototipo	63
6.1.3. Requisitos de datos	63
6.1.4. Requisitos eliminados	64
6.1.5. Estrategias y métodos de elicitación	64
6.1.6. Priorización, validación y negociación de requisitos	65
6.2. Proceso de comparación	66
6.2.1. Valoración de los criterios	67
6.3. Análisis comparativo	68
6.3.1. Justificación de las valoraciones	68
6.4. Sesgos del análisis	71
6.5. Conclusión del análisis comparativo	72

7. Conclusiones	73
A. Anexos prototipo 1	75
A.1. Pantalla de aspirantes	75
A.2. Pantalla de especializaciones para aspirantes	75
A.3. Pantalla de Especialización en Ingeniería de Software para aspirantes	76
A.4. Pantalla de asignaturas de la Especialización en Ingeniería de Software para aspirantes	76
A.5. Pantalla de asignatura específica de la Especialización en Ingeniería de Software para aspirantes	77
A.6. Pantalla de estudiantes	77
A.7. Pantalla de maestrías para estudiantes	78
A.8. Pantalla de Maestría en Ingeniería de Software para estudiantes	78
A.9. Pantalla de asignaturas de la Maestría en Ingeniería de Software para estudiantes	79
A.10. Pantalla de asignatura específica de la Maestría en Ingeniería de Software para estudiantes	79
A.11. Pantalla de login para profesores	80
A.12. Pantalla menú para profesores	80
A.13. Pantalla de cursos para profesores	81
A.14. Pantalla de curso específico para profesores	81
A.15. Pantalla de login para colaboradores	82
A.16. Pantalla menú para colaboradores	82
A.17. Pantalla de crear noticia para colaboradores	83
B. Anexos prototipo 2	85
B.1. Pantalla de inicio	85
B.2. Pantalla de aspirantes	85
B.3. Pantalla de especializaciones para aspirantes	86
B.4. Pantalla de Especialización en Ingeniería de Software para aspirantes	86
B.5. Pantalla de asignaturas de la Especialización en Ingeniería de Software para aspirantes	87
B.6. Pantalla de asignatura específica de la Especialización en Ingeniería de Software para aspirantes	87
B.7. Pantalla de maestrías para estudiantes	88
B.8. Pantalla de Maestría en Ingeniería de Software para estudiantes	88
B.9. Pantalla de asignaturas Maestría en Ingeniería de Software para estudiantes	89
B.10. Pantalla de asignatura específica Maestría en Ingeniería de Software para estudiantes	89
B.11. Pantalla de login para profesores	90
B.12. Pantalla menú para profesores	90
B.13. Pantalla curso específico para profesores	91
B.14. Pantalla de agregar noticia para profesores	91

B.15. Pantalla de login para colaboradores	92
B.16. Pantalla menú para colaboradores	92
B.17. Pantalla de agregar noticia para colaboradores	93
Bibliografía	95

Introducción

El proceso de definición de requisitos es el primer paso dentro de cualquier metodología de desarrollo para un proyecto de software, ya que es la fase durante la cual se especifican los elementos que conformarán al sistema que se desee desarrollar, también es la etapa en la cual se conocen qué funcionalidades tendrá el sistema.

Habiendo dicho lo anterior, la etapa de definición de requisitos es la base para iniciar el proceso de desarrollo de software, por lo tanto, también es la primera etapa para iniciar el desarrollo de una aplicación web.

De una aplicación web se espera un alto grado de usabilidad y amigabilidad con sus usuarios, por lo tanto, actualmente está en auge el desarrollo de sistemas web basado en la experiencia de usuario (UX), lo cual se refiere a realizar el desarrollo teniendo en cuenta posibles interacciones que tengan los usuarios con la aplicación en el futuro y cómo estos se sienten con respecto al manejo de la aplicación.

Pero, por otro lado, las aplicaciones web, normalmente, son desarrolladas mediante metodologías de desarrollo tradicionales, las cuales se fundamentan en la definición de requisitos.

El desarrollo basado en UX también cuenta con una etapa de definición de requisitos, por lo cual se desea determinar qué proceso de definición de requisitos del software es más adecuado (aplicándolo al desarrollo de un sistema web): el proceso de definición de las metodologías tradicionales, o el proceso de definición de las metodologías basadas en UX.

Análisis previo y descripción del problema

1.1. Planteamiento del Problema

A través de los años, el desarrollo de software ha ido evolucionando gracias al surgimiento de distintas metodologías que acompañan todo el proceso de creación de un software. Desde el proceso de definición de requisitos, pasando por el proceso de diseño del software, el de desarrollo y finalmente el proceso de pruebas, los cuales, son a priori, los 4 procesos más importantes a la hora de construir software.

En los últimos años se han utilizado para el proceso de construcción de software, las llamadas metodologías tradicionales de desarrollo de software. Estas se caracterizan por tener un proceso de planeación muy estricto y riguroso, y además, a medida que se avanza en el proceso de construcción, se acompaña con un proceso de documentación exhaustivo y con el cumplimiento estricto del plan realizado durante la etapa inicial del proyecto. [1]

Por otro lado, recientemente ha entrado en auge el uso de metodologías basadas en la experiencia del usuario, también conocidas como metodologías basadas en UX, las cuales, principalmente, plantean un proceso de desarrollo y diseño basado en la experiencia e interacción que tendrán los usuarios con el software a desarrollar. [2]

El desarrollo de aplicaciones web es, actualmente, uno de los campos de desarrollo de software más importantes, ya que el mundo se encuentra en plena era digital, por lo que todos los medios de información y comunicación se han trasladado o han surgido en la web. [3]

El proceso de planeación del desarrollo no solo de una aplicación web, sino también de cualquier software, es de vital importancia, ya que este determina el tiempo que tomará el desarrollo del proyecto, cuál será el costo de la realización del proyecto, y además, con qué metodología se trabajará para tener buenos resultados y optimizar los factores mencionados anteriormente.

Dentro del proceso de planeación, se encuentra una fase vital para el correcto proceso de desarrollo de un proyecto de construcción de software. Dicha fase se conoce como definición de requisitos, la cual se basa en reunirse con el cliente y los interesados en la realización del pro-

yecto para así saber de qué manera quieren que funcione la aplicación y cómo quieren que sea la misma en todos sus aspectos. A su vez, esta fase del desarrollo de software cuenta con una serie de tareas, las cuales se realizan para garantizar una correcta definición de los requisitos del sistema. Dichas tareas son: elicitación, la cual se refiere al proceso de reunirse con el cliente para discutir qué quiere para su sistema; análisis, la cual consiste en realizar un estudio de qué es lo que quiere el cliente para su sistema; verificación, la cual se refiere a verificar con el cliente lo que quiere para el sistema; documentación, la cual se refiere al proceso de documentar lo que el cliente quiere; y finalmente, gestión, la cual se refiere al proceso del manejo de la información que tendrá el sistema. [4]

Todas las metodologías de desarrollo utilizadas actualmente cuentan con una fase de definición de requisitos o algún otro proceso parecido, por lo cual, los dos tipos de metodologías mencionadas anteriormente en este documento, cuentan con este elemento en su proceso de ejecución.

Por todo lo anterior, surgió el interrogante de querer saber qué proceso de definición de requisitos puede ser el mejor para realizar un desarrollo de software, basándose totalmente en los dos tipos de metodologías mencionados anteriormente, y luego, aplicar dichos resultados al caso del desarrollo de una aplicación web, cuyo contenido será todo lo relacionado con los posgrados de la Pontificia Universidad Javeriana Cali y la divulgación de su información.

Para realizar esto, se tuvo en cuenta el trabajo realizado por los estudiantes del curso de Ingeniería de Requisitos, perteneciente al programa de Maestría en Ingeniería de Software de la PUJ Cali. Dicho trabajo se basó en la realización de la definición de requisitos de la aplicación ya mencionada, aplicando de manera rigurosa la Ingeniería de Requisitos.

Para el otro lado del análisis, se realizó una investigación sobre el proceso de definición de requisitos en metodologías UX, para así poder realizar las comparaciones que sean pertinentes.

Teniendo en cuenta esa información, se quiso saber qué proceso de definición de requisitos es más adecuado para el desarrollo: la tradicional o la basada en UX para el desarrollo de una aplicación web.

1.1.1. Formulación

¿Qué proceso de definición de requisitos es más efectivo para el desarrollo de una página web?

1.1.2. Sistematización

- ¿Qué estrategias de definición de requisitos se emplean en las metodologías basadas en UX?

- ¿Cómo se pueden definir los requisitos de la aplicación web, teniendo en cuenta las estrategias de definición de requisitos para metodologías basadas en UX?
- ¿De qué manera se puede realizar un análisis comparativo ambos procesos de definición de requisitos?
- ¿De qué manera se puede seleccionar cuál de los dos procesos de definición de requisitos es el más eficaz para el web?
- ¿Cómo se puede desarrollar el prototipo web, basándose en los resultados obtenidos en la comparación de los análisis de ambos procesos de definición de requisitos?

1.2. Objetivos

1.2.1. Objetivo General

Realizar un análisis comparativo del proceso de definición de requisitos de una metodología de desarrollo tradicional y una metodología UX para el desarrollo de una aplicación web, para así determinar cuál es más adecuada para el desarrollo web.

1.2.2. Objetivos Específicos

- Identificar las estrategias de definición de requisitos en las metodologías basadas en UX.
- Definir los requisitos de la aplicación web por medio de las estrategias de definición de requisitos para metodologías basadas en UX.
- Realizar un análisis comparativo de los procesos de definición de requisitos entre el proceso realizado con metodologías UX y un proceso realizado de manera rigurosa aplicando la Ingeniería de Requisitos.
- Identificar cuál de los dos metodologías es más eficaz para el desarrollo web.
- Desarrollar el prototipo web basado en los resultados obtenidos en el análisis comparativo de ambos procesos de definición de requisitos.

1.3. Justificación

La realización de este proyecto se basó, principalmente, en la oportunidad que pueden brindar las metodologías UX para el proceso de definición de requisitos del desarrollo web, lo cual puede permitir que se exploren más opciones a la hora de seleccionar una metodología de desarrollo de software diferente a las metodologías tradicionales, lo que podría conllevar consigo mejores resultados en los procesos de desarrollo.

1.4. Marco de Referencia

1.4.1. Áreas Temáticas

- Software - Ingeniería de software - Desarrollo de software - Metodologías de desarrollo de software - Ingeniería de requisitos.
- Software - Ingeniería de software - Desarrollo de software - Desarrollo de aplicaciones web.
- Software - Ingeniería de software - Desarrollo de software - Metodologías de desarrollo de software - Metodologías tradicionales de desarrollo de software.
- Software - Ingeniería de software - Desarrollo de software - Metodologías de desarrollo de software - Metodologías de desarrollo de software ágiles.
- Software - Ingeniería de software - Desarrollo de software - Metodologías de desarrollo de software - Metodologías de desarrollo de software basadas en el experiencia de usuario (UX).

1.4.2. Marco Teórico

1.4.2.1. Ingeniería de requisitos

La ingeniería de requisitos suele ser el primer proceso que se realiza en la etapa de desarrollo de software. Según Pohl y Rupp[4], la ingeniería de requisitos es un acercamiento sistemático y disciplinado para la especificación y el manejo de los requisitos teniendo en cuenta dos grandes objetivos: conocer qué requisitos son relevantes y entender los deseos y necesidades del cliente (también conocido como Stakeholder). Esto con el propósito de manejar y administrar de una manera eficaz la información brindada por los requisitos y poder clasificarlos de acuerdo a su nivel de importancia para el desarrollo del proyecto.

La ingeniería de requisitos está conformada por un conjunto de tareas, las cuales facilitan el proceso de definición de los requisitos. Dichas tareas son:

- Elicitación o levantamiento de requisitos: proceso de interactuar con los stakeholders para discutir los requisitos del sistema.
- Análisis de requisitos: proceso de analizar los requisitos obtenidos en el proceso de elicitación.
- Verificación de los requisitos: proceso de verificar los requisitos con los stakeholders de acuerdo a los resultados obtenidos en la etapa de análisis.
- Documentación de los requisitos: proceso de documentar los requisitos obtenidos de los procesos anteriores.

- Gestión de los requisitos: proceso de manejar y gestionar la información de los requisitos documentados.

1.4.2.2. Experiencia de usuario

Según Helena Tokkonen y Pertti Saariluoma [5], la experiencia de usuario (UX) se refiere a un acercamiento holístico, incluyendo el contexto en el que se usa, emociones que genera y significado de un producto, también a la relación producto consumidor y a cómo se entiende que un producto debe funcionar. Lo anterior quiere decir que, la experiencia de usuario principalmente es todo contacto y acercamiento que tenga un usuario con un producto, y todo lo que esta interacción pueda generarle a la persona.

1.4.2.3. Diseño de experiencia de usuario

De acuerdo a un artículo de Platzi, una de las web de cursos virtuales de Latinoamérica[6], el diseño de experiencia de usuario (UXD), se basa principalmente en la tarea de estudiar a los usuarios, sus interacciones con los diferentes sistemas y el impacto que estos sistemas, ya sean producto o servicios, puedan tener en su entorno o en su vida. Es decir, el principal objetivo del UXD es entender los deseos y necesidades de potenciales usuarios, para así poder brindarles experiencias únicas y satisfactorias, en lo que al uso del producto o servicio se refiere, teniendo en cuenta los objetivos de negocio y los objetivos de los propios usuarios.

1.4.2.4. Usabilidad

Según el compendio de normas ISO 25000 [7], la usabilidad se refiere a la capacidad que tiene un software para ser entendido, aprendida, utilizado y llamativo para los usuarios a los que va dirigido. Se puede entender también como qué tan acorde y adaptado el uso de un software para los usuarios finales a los cuales va dirigido, que pueden variar ya sea por grupos de edad, geográficos, entre otros. Las normas ISO 25000 también plantean que, la usabilidad, siendo una característica del software, trae consigo una serie de subcaracterísticas, las cuales son las siguientes:

- **Capacidad para reconocer su adecuación:** Capacidad del software que permite a los usuarios saber si el mismo se adecua a sus necesidades.
- **Capacidad de aprendizaje:** Capacidad del software que permite a los usuarios aprender a interactuar con el y sus funcionalidades.
- **Capacidad para ser usado:** Capacidad del software que permite a los usuarios operar y utilizar el mismo con facilidad.

- **Protección contra errores de usuario:** Capacidad del software que permite proteger y evitar que los usuarios realicen errores a la hora de su uso.
- **Estética de la interfaz de usuario:** Capacidad de la interfaz de usuario del software de ser agradable a la vista de los usuarios.
- **Accesibilidad:** Capacidad del software que le permite ser utilizado por distintos tipos de usuarios.

1.4.2.5. Metodologías de desarrollo tradicionales

Las metodologías de desarrollo de software son procesos que guían todo el proceso de desarrollo. Según Leo R. Vijayarathy y Charles W. Butler[8], las metodologías de desarrollo de software proveen un framework o plantilla para los procesos de planeación, ejecución y administración del software. Ahora, las metodologías tradicionales, son las que llevan mucho tiempo siendo utilizadas, y hasta el día de hoy continúan en vigencia. Metodologías tales como RUP, cascada, espiral, entre otras, han sido las encargadas de guiar el desarrollo de proyectos de software desde hace ya varios años.

1.4.2.6. Metodologías de desarrollo ágiles

Las metodologías de desarrollo ágiles [9] se refieren a metodologías de desarrollo de software basadas en el desarrollo iterativo e incremental, procesos en los cuales los requisitos y las soluciones evolucionan sobre la marcha de la realización del proyecto, esto como respuesta a las necesidades que vayan surgiendo durante este proceso, lo cual, también los hace procesos adaptativos.

Cada iteración de estas metodologías está regida por los procesos de planeación, análisis de requisitos, diseño, desarrollo, pruebas y documentación. Con cada una de las iteraciones se busca entregar y/o realizar pequeñas nuevas funcionalidades del sistema o producto que se este desarrollando.

Son metodologías que priorizan la comunicación entre miembros del equipo de desarrollo antes que la documentación, lo cual también es una de las causantes de que este tipo de metodologías de desarrollo optimicen los tiempos de ejecución de los proyectos. Lo anterior no significa que no se realice ningún tipo de documentación, solamente que esta se realiza solo cuando es estrictamente necesario.

Entre las más utilizadas se encuentran SCRUM, eXtreme Programming (XP), Agile Unified Process (AUP), Lean Software Development (Lean Startup), entre otras.

1.4.3. Trabajos Relacionados

Se encontraron dos trabajos que tratan temáticas similares a las abordadas en este proyecto.

En primer lugar, *An introduction to experience requirements*[10] habla sobre el proceso de elicitación, captura y representación de los requisitos de un sistema a partir de la experiencia de usuario, proceso del cual, mediante el uso de un modelo de estímulo, percepción y respuesta, se obtienen los requisitos de experiencia, termino que se refiere a como la descripción de las experiencias de usuario al momento de hacer contacto con un prototipo de la aplicación (experiencias funcionales) y también abarca lo que tiene que ver con la satisfacción en cuanto al uso de la aplicación (experiencias no funcionales). Todo lo anterior es aplicado a los videojuegos.

Por otro lado, en *Integrating requirements engineering and user experience design in Product life cycle Management*[11] plantea el gran avance que permitió el uso de metodologías tradicionales de desarrollo de software, tales como Cascada, Espiral, RUP, entre otras, a la evolución del desarrollo de software, sin embargo, estas se quedan cortas al momento de medir la calidad, ya que no suelen medir niveles de interacción de usuario y de usabilidad. Lo anterior trae a la mesa el UXD (Diseño de experiencia de usuario), el cual se ajusta al proceso de desarrollo pensando siempre en hacer el producto usable y amigable. Finalmente plantean la importancia de incluir el UXD y la ingeniería de requisitos durante todo el ciclo de vida de los procesos de desarrollo para así lograr un mejor producto.

Desarrollo del Proyecto

2.1. Selección de la metodología de desarrollo

Para el proceso de selección de la metodología de ingeniería de requisitos de este proyecto se tuvo en cuenta un proceso de calificación numérica, donde la metodología que obtuvo el valor más alto, fue la seleccionada para la realización de todo los procesos del proyecto.

2.1.1. Condiciones previas

Para evaluar las condiciones previas para la selección de una metodología de ingeniería de requisitos para la realización del prototipo de la aplicación web de este proyecto, se utilizó un rango de valores enteros de 1 a 5, cuyo valor se entiende de la siguiente manera:

- 1: No es nada importante para la realización este proyecto.
- 2: Es poco importante para la realización de este proyecto.
- 3: Puede ser importante para la realización de este proyecto.
- 4: Es importante para la realización de este proyecto.
- 5: Es muy importante para la realización de este proyecto.

A continuación, se procede a realizar el proceso de valoración de cada una de las condiciones previas, las cuales fueron definidas en el siguiente cuadro:

Proceso	Característica	Valor
Tamaño del proyecto (1)	El objetivo del proceso de desarrollo del proyecto, es realizar un prototipo web, por lo cual se puede decir que es un proyecto mediano/pequeño.	3
Interacción (2)	Al utilizar metodologías centradas en la experiencia de usuario, la interacción con stakeholders y usuarios es muy importante.	5
Tiempo (3)	Se tienen aproximadamente 3 meses para la realizar el proyecto, lo cual indica que se necesita optimizar el tiempo del mismo.	4
Disponibilidad de stakeholders (4)	Se requiere una comunicación continua y eficaz con los stakeholders, ya que se quiere mejorar su experiencia con el prototipo de la aplicación web.	5
Personal (5)	El proyecto involucra solo a una persona. Igualmente se tiene el apoyo del director de carrera como consejero y guía.	2
Desarrollo de software (6)	Es mejor si las metodologías son muy afines con el proceso de desarrollo de software y el background del personal.	3

Cuadro 2.1: Condiciones Previas.

Después de la valoración, la cual fue validada junto con el director del trabajo de grado, se decidió marcar cada una de las condiciones como un número, el cual se utilizó en la parte de los criterios, para relacionar cada criterio con una de estas condiciones, y así determinar la importancia de cada criterio dentro de la realización del proyecto.

2.1.2. Criterios de evaluación

Luego de realizar la verificación de los criterios y la búsqueda de literatura al respecto de metodologías UX, se encontró que las metodologías mejor documentadas, más usadas y las que más se acoplan al desarrollo de este proyecto son las siguientes:

- Design Thinking:** según Peter Denning en su artículo *Design Thinking* [12], Design Thinking es un enfoque innovativo para el diseño de nuevos productos y/o soluciones. Este nuevo enfoque de diseño fue desarrollado por la compañía IDEO y su fundador David Kelly.

Este enfoque o metodología de diseño de producto se basa principalmente en sumergirse alrededor de las preocupaciones, intereses y valores de los posibles usuarios del producto o servicio que se está diseñando o desarrollando. En pocas palabras, Design Thinking se

enfoca en todo tipo de interacción o sentimiento que le pueda generar la posible utilización del producto que se desea desarrollar, esto con el objetivo de ofrecer una experiencia de usuario completamente satisfactoria y agradable para potenciales usuarios.

Adicionalmente, Dobrigkeit y de Paula plantean en *Design Thinking in Practice: Understanding Manifestations of Design Thinking in Software Engineering* [13], que Design Thinking es un concepto que promete una innovación incremental durante todo el proceso de diseño de producto, utilizando un enfoque más centrado en el usuario para el proceso de innovación. Ambas definiciones plantean que el principal enfoque de Design Thinking es el diseño centrado en el usuario pero también plantean lo fácil que se puede adaptar específicamente al desarrollo de software. Para realizar dicho proceso de adaptación, se necesita la utilización de alguna metodología de desarrollo de software, preferiblemente una metodología ágil, ya que, tanto estas como el Design Thinking se basan en el desarrollo iterativo e incremental. La combinación de ambos elementos da lugar a un proceso similar a Agile UX.

- **Agile UX:** las metodologías de desarrollo de software ágil [9], como se mencionó anteriormente, son procesos de desarrollo que se basan en el desarrollo incremental e iterativo, es decir, se divide el proceso total en varias etapas, separadas ya sea por tiempo o por logros. En cada una de estas etapas se desarrollan y prueban una serie de funcionalidades, definidas anteriormente en los requisitos del proyecto. En cada etapa se van agregando más elementos al producto final, de ahí lo de desarrollo incremental. Por su lado, el diseño centrado experiencia de usuario (UXD) [2], como su nombre lo indica, se refiere a un proceso de diseño basado en todo tipo de interacciones, experiencias, sentimientos que pueda generar el producto que se este desarrollando en los usuarios hacia los cuales va dirigido. Teniendo en cuenta los dos conceptos anteriores, Agile UX [14] nace como la integración y adaptación mutua de ambos proceso, ya que el diseño centrado en el usuario también es un proceso incremental e iterativo así como lo son las metodologías de desarrollo ágil. Debido a que ambos proceso se basan en paradigma similar, el proceso de integración de ambos es relativamente sencillo. Se suelen integrar el proceso de diseño a las diferentes etapas del proceso de desarrollo.
Para Agile UX se suele usar como base la metodología SCRUM [15], proceso que cuenta con un conjunto de etapas, las cuales se puede adaptar fácilmente el diseño centrado en el usuario. Por ejemplo, en el desarrollo de los backlogs (proceso de definición de requisitos e historias de usuarios) generalmente se centra en adquirir requisitos funcionales del sistema que se quiere desarrollar, y con su transformación a Agile UX, también se adquirirían requisitos de interacción que debería tener el sistema para brindar una buena experiencia de usuario [16].
- **Lean UX:** como lo plantea Tatiana Batova en *Extended Abstract: Lean UX and Innovation in Teaching* [17], Lean UX es una proceso que combina elementos de Lean Startup, Design Thinking, experiencia de usuario y metodologías de desarrollo ágil. En primer lugar, Batova dice que Lean Startup es un proceso de diseño y desarrollo de producto incremental, y lo que lo diferencia de otros procesos regidos bajo este paradigma, es que se busca desarrollar

un producto que tenga demanda en el mercado, un producto que cubra una necesidad existente de un grupo de personas al cual quiera ser dirigido. Lean Startup busca el desarrollo de un producto mínimo viable (MVP). Este tarea se logra con el apoyo de los elementos brindados por los otros procesos mencionados anteriormente. Toma elementos del desarrollo iterativo e incremental de las metodologías ágiles y todo el proceso de desarrollo es acompañado por procesos de diseño centrado en el usuario (Design Thinking, UX), para así garantizar, la completa satisfacción de la necesidad que desea ser cubierta con el desarrollo del producto. Esto también es abordado en *Lean UX - The Next Generation of User-Centered Agile Development?* [18], donde sus autores mencionan que Lean UX es una combinación de las filosofías del desarrollo ágil, Design Thinking y Lean Startup.

Después de leer e investigar sobre estas metodologías, se establecieron unos criterios, en conjunto con el director del trabajo, que deben cumplir las mismas para realizar la selección de la que se utilizó en el desarrollo de este proyecto. Dichos criterios se describen a continuación:

- Conocimiento previo: se refiere al conocimiento que ya posee el equipo a cerca de la utilización de las metodologías.
- Curva de aprendizaje: se refiere a la facilidad de aprendizaje con respecto al tiempo de cada una de las metodologías.
- Requisitos funcionales: se refiere a que tanto la metodología abarca la recolección y definición de requisitos funcionales.
- Requisitos no funcionales: se refiere a que tanto la metodología abarca la recolección y definición de requisitos no funcionales.
- Prototipos: se refiere a la facilidad que brinda la metodología de realizar prototipos de la aplicación durante todo el proceso de desarrollo del proyecto.
- Requisitos no conocidos: se refiere a la capacidad que brindan las metodologías de encontrar nuevos requisitos durante el desarrollo del proyecto.
- Tiempo: se refiere a que tanto la metodología se adecua bien al tiempo estimado de realización del proyecto.
- Reuniones: se refiere a que tanto la metodología abarca reuniones con los clientes y los stakeholders.
- Tamaño del proyecto: se refiere a que tanto se acoplan las metodologías con el tamaño del proyecto.
- Enfocado al desarrollo de software: se refiere a que tan relacionadas están las metodologías con el desarrollo de software, ya que son utilizadas, mayoritariamente, en proyectos de diseño y desarrollo de productos en general.

- Etapas de desarrollo de software: se refiere a que tanto se acoplan las metodologías con todas las etapas del proceso de desarrollo de software (ingeniería de requisitos, diseño, desarrollo, pruebas).

- Tamaño del equipo: se refiere a que tanto se adecuan las metodologías al tamaño del equipo encargado de la realización del proyecto.

- Costo: refiere a que tanto se pueden adecuar las metodologías con respecto al costo del proyecto.

- Usabilidad: se refiere a que tanto están enfocadas las metodologías con respecto a temas de usabilidad.

Ahora, el siguiente paso fue calificar cada uno de los criterios con respecto a cada metodología, para así seleccionar la más adecuada para este proyecto. En esta proceso de clasificación, se relacionó cada uno de los criterios con una de las condiciones previas definidas arriba. Dicha relación se hace de acuerdo a la afinidad que tenga un criterio con una de las condiciones previas. A continuación, se realiza el proceso de calificación:

Metodologías/ Aspectos	Grupo	Peso	Design Thinking	Agile UX	Lean UX
Conocimiento previo	6	3	4	5	1
Curva de apren- dizaje	6	3	5	5	3
Requisitos fun- cionales	2	5	4	5	5
Requisitos no funcionales	2	5	2	2	2
Prototipos	2	5	5	5	5
Requisitos no co- nocidos	2	5	5	5	5
Tiempo	3	5	5	4	5
Reuniones	4	5	5	5	5
Tamaño del pro- yecto	1	3	5	5	3
Enfocado al desa- rrollo de software	6	3	3	5	5
Etapas de desa- rrollo de software	6	3	3	5	5
Tamaño del equi- po	5	2	3	3	2
Costo	1	3	5	5	5
Usabilidad	2	5	5	5	5

Cuadro 2.2: Criterios de evaluación.

Después de realizar el proceso de clasificación de cada uno de los criterios para cada una de las metodologías, teniendo en cuenta la validación pro parte del director del trabajo, se procedió a realizar el proceso de recolección de puntaje, proceso en el cual se multiplican las calificaciones que se le dieron a cada uno de los criterios dependiendo de cada metodologías, el peso de cada criterio, el cual está definido por el grupo al que pertenece cada uno de estos. A continuación se realizó dicho proceso:

Metodologías/ Aspectos	Design Thinking	Agile UX	Lean UX
Conocimiento previo	12	15	3
Curva de aprendi- zaje	15	15	9
Requisitos fun- cionales	20	25	25
Requisitos no funcionales	10	10	10
Prototipos	25	25	25
Requisitos no co- nocidos	25	25	25
Tiempo	25	20	25
Reuniones	25	25	25
Tamaño del pro- yecto	15	15	9
Enfocado al desa- rrollo de software	9	15	15
Etapas de desa- rrollo de software	9	15	15
Tamaño del equi- po	6	6	4
Costo	15	15	15
Usabilidad	25	25	25
Total	236	251	230

Cuadro 2.3: Puntuaciones finales.

Se puede observar que los valores totales de cada una de las 3 metodologías no son muy lejanos entre sí, lo cual puede ser un indicio de que, al menos en este proyecto, se podría trabajar con cualquiera de las 3 metodologías. Pero, como este ejercicio se realizó con la finalidad de poder seleccionar la metodologías más acorde para lo que necesita el proyecto, se puede observar que Agile UX fue la metodología mejor valorada, lo cual indica que con esta misma se trabajará de ahora en adelante en la realización del proyecto, en el proceso de ingeniería de requisitos y en el proceso de desarrollo del software como tal.

Metodología seleccionada: Agile UX

3.1. Descripción general

Como se mencionó en el final del capítulo anterior, la metodología de desarrollo seleccionada para el desarrollo y acompañamiento de este proceso es Agile UX.

Agile UX [16] es una metodología de desarrollo de software que combina elementos de las metodologías de desarrollo ágil y los procesos de diseño centrados en el usuario. Ambos procesos tienen una gran facilidad de integración entre, ya que ambos son procesos o metodologías de carácter incremental e iterativo [2][9]. Que sean incrementales quiere decir que a medida que avanza la realización del proyecto o producto, se van agregando más funcionalidades y/o características. Ahora, que sea iterativo, se refiere a que el proceso se divide en un determinado número de etapas, en las cuales se van desarrollando funcionalidades y/o características previamente definidas.

La metodología de desarrollo ágil base de Agile UX, es este caso, es SCRUM, ya que su proceso iterativo (por medio de sprints) [15], facilita el proceso de integración con el diseño centrado en el usuario, debido a que en cada uno de los sprints, se añaden elementos del proceso de diseño centrado en el usuario. Esta integración será detallada más adelante.

En algunas fases o etapas de la metodología, se generan ciertos elementos o artefactos, los cuales sirven ya sea para complementar el proceso, o para recolectar la evidencia del proceso que se está realizando.

3.2. Proceso detallado aplicado al proyecto

3.2.1. Reuniones con stakeholders

Como en el desarrollo de cualquier proyecto, sin importar por cuál metodología o proceso se rija, el primer paso que se da es reunirse con las partes interesadas en la ejecución del proyecto. Estas personas o partes interesadas se conocen como Stakeholders [19]. Los stakeholders suelen ser los jefes de la empresa para la cual se va a desarrollar el proyecto, personas cercanas a él en el ambiente de su compañía, personas de interés para el proyecto como los futuros usuarios, entre otras tantas.

El rol de los stakeholders principalmente es dar a conocer que es lo que quieren con respecto al proyecto, es decir, qué quieren que haga el proyecto, qué funcionalidades esperan que tenga, etc.

En el caso de este proyecto, los stakeholders fueron:

- Juan Carlos Martínez (Stakeholder principal, Director de Posgrados de la Facultad de Ingeniería y Ciencias de la Pontificia Universidad Javeriana Cali)
- Maria Contanza Pabón (Profesora del Departamento de Electrónica y Ciencias de la Computación de la Facultad de Ingeniería y Ciencias de la Pontificia Universidad Javeriana Cali)
- Colaboradores área de posgrado
- Algunos profesores de posgrado
- Algunos estudiantes de posgrado
- Algunos estudiantes de pregrado, los cuales serán vistos también como aspirantes

3.2.2. Historias de usuario

En el proceso inmediatamente anterior, se discute con los stakeholders las funcionalidades y aspectos que quieren que conformen al sistema. Dichos elementos son conocidos como requisitos. Estos requisitos son obtenidos, moldeados y definidos, mediante un proceso estructurado, que puede involucrar prácticas como prototipado, conversaciones, entre otras.

Dichos requisitos se relacionan a aspectos ya sean funcionales o de interacción de la aplicación, en ocasiones a ambos. Este proceso se da gracias al uso de Agile UX, ya que permite obtener requisitos de ambos tipos, para asegurarse una definición más detallada de los elementos que conformarán al sistema. Este proceso se detallará más adelante en este documento, más exactamente, al momento de aplicar la metodología con los respectivos stakeholders del proyecto.

Estos requisitos se agrupan, dependiendo de posibles relaciones o dependencias entre si, en historias de usuario. Las historias de usuario [20] son un elemento o artefacto, en el cual se plasman los requisitos obtenidos en los encuentros con los stakeholders. En las historias de usuario, generalmente se ponen uno o más requisitos, relacionados entre si.

3.2.3. Backlog funcionalidades

El paso siguiente a la definición de historias de usuario, es la definición del backlog (en el caso de este proyecto, son tres). Un backlog [21] es una pila o acumulación de tareas pendientes por realizar. En este caso, dichas tareas serían representadas por las historias de usuario. Estas

tareas se priorizan de acuerdo a su importancia en el funcionamiento del proyecto.

El primer backlog entonces, es el backlog de funcionalidades, espacio en el cual se almacenarán todas las historias de usuario que estén relacionadas a requisitos funcionales del sistema. Como se mencionó anteriormente, estas tareas (historias de usuarios), serán priorizadas de acuerdo a su importancia en el aspecto funcional del sistema que se va a desarrollar.

El artefacto obtenido en esta fase, es el conjunto de las historias de usuario de las funcionalidades del sistema.

3.2.4. Backlog interacciones

De la misma forma que el primer backlog, se define el segundo. En este caso, este es el backlog de interacciones, el cual se refiere al conjunto de todas las historias de usuario relacionadas con temas de interacción que tendrá el sistema, incluyendo también los aspectos visuales de la aplicación. Las historias de usuarios pertenecientes a este conjunto también son priorizadas.

El artefacto obtenido en esta fase, es el conjunto de las historias de usuario de las interacciones del sistema.

3.2.5. Backlog unificado

Finalmente, está el backlog unificado, el cual representa el conjunto de historias de usuario del backlog de funcionalidades con las historias de usuario del backlog de interacciones, es decir la relación de aspectos visuales o interactivos que dependen o están ligados de algún modo a aspectos funcionales del sistema. Este conjunto de nuevas historias de usuario, será la base para la planeación (sprint backlog) y ejecución del proceso (sprints) de desarrollo. Una vez más, las historias de usuario son priorizadas.

El artefacto obtenido en esta fase, es el conjunto de las historias de usuario relacionadas (funcionales-interacciones) del sistema.

3.2.6. Sprint backlog

Después de la realización del backlog unificado, se comienza el proceso de planeación de cada una de las iteraciones. Se recopilan cierto número de historias de usuario, de acuerdo a su nivel de prioridad y su dependencia entre sí, para de esta forma empezar la ejecución del sprint.

En cada sprint backlog puede variar el número de historias de usuario asignadas, dependiendo de factores como complejidad de realización, tiempo de realización, cantidad de dependencias, entre otras.

El proceso de realización de sprint backlog, se hace siempre antes del inicio de cada sprint, para así definir de manera clara qué tareas se van a manejar en el mismo.

Como este proceso es iterativo, en cada de las iteraciones se obtienen un artefacto, el cual es el conjunto de todas las historias de usuario que fueron seleccionadas y asignada a cada uno de los sprints.

3.2.7. Sprint

Como se dijo antes, un sprint es cada una de las iteraciones del proyecto. En cada un de estos sprints, se define un sprint backlog, es decir, el conjunto de tareas que se van a realizar durante esta iteraciones. Para cuestiones de este proyecto, cada uno de los sprint, tiene una duración de una semana. Durante esta semana de trabajo se realizan las tareas definidas en el sprint backlog, ya sean tareas de funcionalides o de interacciones. Estas tareas son realizadas por el equipo del proyecto y son vigiladas por el SCRUM Master, que para efectos del proyecto, son la misma persona.

3.2.8. Reuniones para feedback

Durante cada sprint, se realizan pequeñas reuniones con los stakeholders, con el objetivo de mostrar los avances en la realización de las tareas definidas en el sprint backlog. Estas reuniones son muy útiles para el desarrollo del proyecto, ya que permiten obtener retroalimentación del proceso que se está realizando, lo cual puede dar pie a que surjan nuevas sugerencias sobre funcionalidades o interacciones que los stakeholders sientan que hacen falta para que el sistema sea más completo. Estas sugerencias, son discutidas y analizadas por parte del equipo, para ver si son acertadas y viables de implementar.

En estas reuniones se exponen prototipos funcionales, para que los stakeholders pueden interactuar con ellos, y así dar la retroalimentación para saber que aspectos de funcionalidad e interacción se deben mejorar o se deben eliminar.

El artefacto obtenido en estas reuniones, son nuevas historias de usuario, las cuales pueden ser originadas por funcionalidades o interacciones que le faltan al sistema, o por simples correcciones que deben ser solucionadas en el mismo.

3.2.9. Pruebas funcionales y de interacción

Después de la finalización de cada sprint, el paso siguiente son las pruebas. Este proceso también incluye a los stakeholders, ya que se realiza un proceso similar al de las reuniones para feedback, con la pequeña diferencia, que esta vez se espera probar el resultado de todas las tareas finalizadas, definidas previamente en el sprint backlog.

El principal objetivo de este proceso, es encontrar bug, errores o problemas en el funcionamiento de la aplicación hasta el punto que se haya desarrollado, esto en el ámbito de lo funcional.

Por otro lado, en el ámbito interactivo, se busca recolectar información acerca de que tan cómodo y usable es el sistema, además de también buscar algún error en el ámbito de su usabilidad.

En esta fase, el artefacto obtenido son datos del funcionamiento y la interacción del sistema. También se pueden obtener nuevas historias de usuario, relacionadas a fallas o elementos faltantes dentro del proceso de ejecución del sistema o su ámbito interactivo.

3.3. Diagrama de la metodología

El siguiente diagrama, resume el proceso mencionado anteriormente y lo ajusta a los diferentes actores participantes en el desarrollo de este proyecto:

Un detalle a tener en cuenta en el diagrama presentado en la figura 3.1, es que, como la metodología es un proceso iterativo, se realizan varias iteraciones, en este caso, sprints, y además, también se señala los pasos a seguir luego de la finalización de la última iteración del proceso.

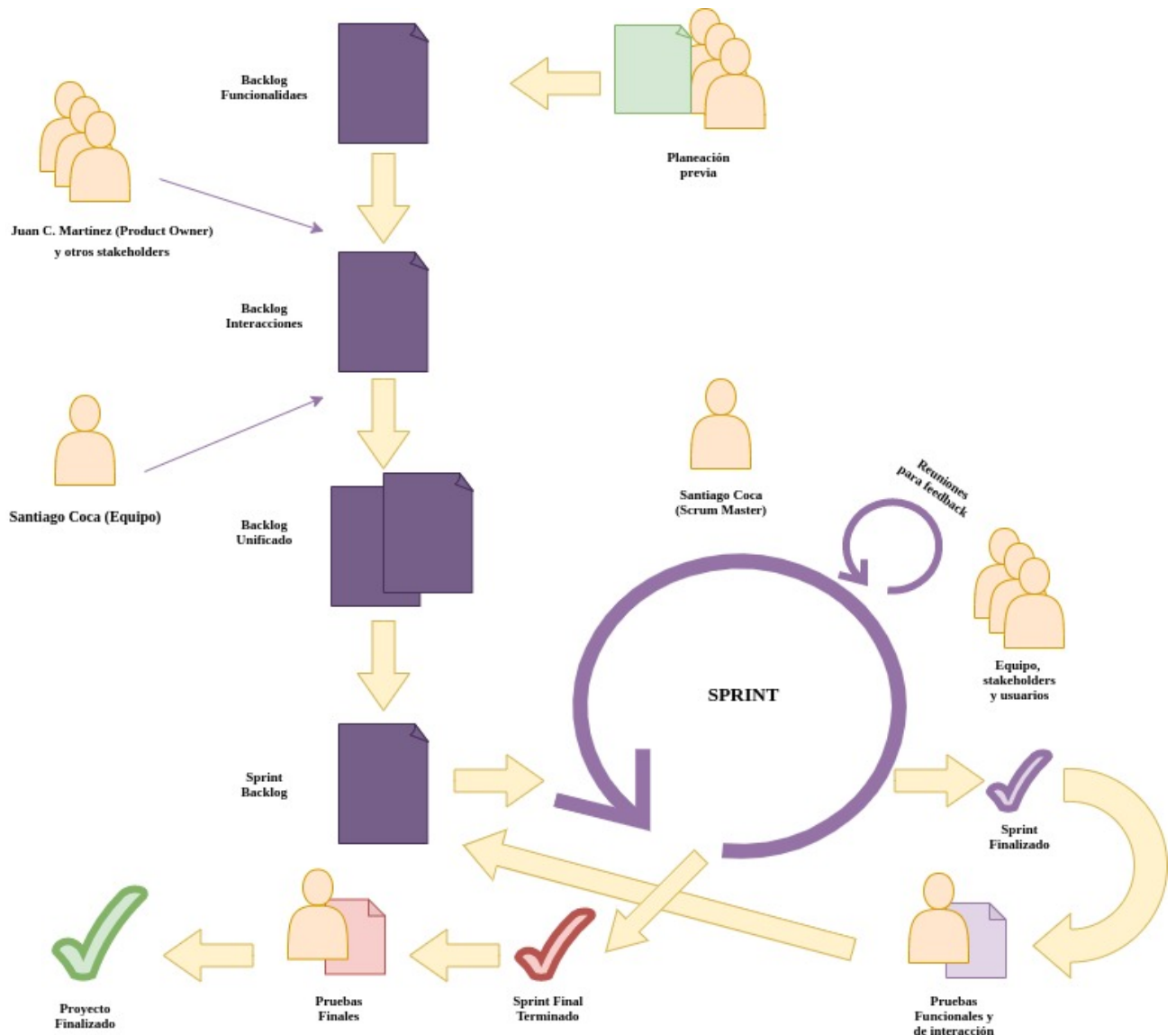


Figura 3.1: Diagrama de la metodología Agile UX

Elicitación de Funcionalidades e Interacciones

4.1. Tipos de usuarios

El primer paso para realizar los procesos de elicitación de requisitos funcionales y de interacciones, es la definición del tipo de usuarios a los cuales va enfocada la aplicación web, de esta manera, se optimiza el proceso de elicitación, ya que los requisitos obtenidos se pueden clasificar de una manera más detallada y sencilla. A continuación, se define cada uno de los grupos de usuarios:

4.1.1. Aspirantes

Los aspirantes son aquellas personas que están interesadas en cursar algún programa de posgrado de la Facultad de Ingeniería y Ciencias de la Pontificia Universidad Javeriana Cali. Estos a su vez, pueden ser estudiantes de pregrado de la misma institución o también ajenos a la misma.

4.1.2. Estudiantes

Los estudiantes son aquellas personas pertenecientes a algún programa de posgrado de la Facultad de Ingeniería y Ciencias de la Pontificia Universidad Javeriana Cali.

4.1.3. Docentes

Los docentes son aquellas personas que dictar las asignaturas pertenecientes a los programas de posgrado de la Facultad de Ingeniería y Ciencias de la Pontificia Universidad Javeriana Cali.

4.1.4. Personal administrativo y colaboradores

El personal administrativo y los colaboradores son las personas encargadas de los aspectos administrativos dentro de la Facultad de Ingeniería y Ciencias de la Pontificia Universidad Javeriana Cali. En este conjunto entran personas como los directores de los distintos programas de posgrados, asistentes y secretarías de la facultad.

Después de realizar la definición de cada un de los tipos de usuario, y describir cuales son sus funciones y/o intereses, el siguiente paso fue detallar los métodos y herramientas que se emplearon para realizar ambos procesos de elicitación. Por otro lado, se debe tener en cuenta que, todas las personas seleccionadas para el proceso de entrevistas, hacen parte de alguna forma de la Pontificia Universidad Javeriana Cali, lo que puede generar una serie de sesgos con respecto a aspectos visuales o funcionalidades que puedan ser similares a las aplicaciones oficiales de la misma universidad.

4.2. Funcionalidades

En primer lugar, el primer proceso de elicitación de funcionalidades consistió en una reunión con el principal stakeholder de la aplicación, el director de posgrados de la Facultad de Ingeniería y Ciencias de la Pontificia Universidad Javeriana Cali, Juan Carlos Martinez Arias, reunión en la cual se discutió lo que el desea que tenga la aplicación, en cuestión de funcionalidades enfocadas en cada tipo de usuario previamente definido.

De esa reunión se pudieron extraer las siguientes funcionalidades o requisitos, de acuerdo a lo que quiere el stakeholder:

- El sistema debe incluir información de los programas de posgrado de la facultad de Ingeniería de la Pontificia Universidad Javeriana Cali.
- La información presentada por el sistema debe incluir información de asignaturas, programas académicos, eventos y novedades de la facultad
- El sistema debe estar enfocado a cuatro tipos de usuarios diferentes: aspirantes, estudiantes, profesores y personal administrativo o colaboradores.
- El sistema debe tener la información organizada de acuerdo a los cuatro tipos de usuarios que maneja.
- El sistema debe ser una aplicación web.

Con el contenido obtenido en dicha reunión, el paso siguiente consistió en realizar entrevistas a otros stakeholders y posibles usuarios, pertenecientes a los grupos definidos anteriormente.

Acto seguido, para realizar un proceso elicitación de las funcionalidades de la aplicación más detallado, se definió un formato de entrevista para cada tipo de usuario, es decir, cuatro formatos en total. Estos formatos comparten elementos, pero lo que los diferencia entre si, son una serie de preguntas que van dirigidas hacia cada grupo en específico.

La estructura de la entrevista fue la siguiente:

- Consta de una sección informativa, en la cual se explica el objetivo de la aplicación y las funcionalidades y elementos que desea el director de posgrados de la Facultad de Ingeniería y Ciencias, elementos obtenidos en la reunión que se expone anteriormente.
- Por otro lado, cuenta con una sección explicativa del por qué se realiza dicha entrevista.
- Finalmente, están las preguntas, las cuales varían dependiendo del usuario al que van enfocadas, a excepción de algunas que son generales en los cuatro formatos

A continuación, se muestra el formato utilizado para realizar las entrevistas:

Formato Entrevista
<p>El director de los programas de posgrado, Juan Carlos Martínez, plantea que, en los distintos portales académicos, no hay información suficiente que pueda necesitar una persona que aspire a realizar un posgrado perteneciente a la facultad de ingeniería y ciencias de la universidad (PUJ). Teniendo esto en cuenta, la solución planteada por el director se basa en la realización de una aplicación web, en la cual se distribuya y se organice la información de los programas de posgrado de una manera más detallada para que las personas interesadas puedan verificar dicha información y así puedan decidir enlistarse en los distintos programas. Adicionalmente, también plantea que esta aplicación sea orientada al uso de estudiantes pertenecientes a los distintos programa, profesores de los mismos, y a algunos miembros de la parte administrativa de este sector de la facultad.</p> <p>Teniendo en cuenta lo anterior, a continuación se realizará una pequeña entrevista, con el objetivo de verificar, adicionar y/o cambiar, algunos de los aspectos y/o funcionalidades planteadas por el profesor Juan Carlos Martínez, en una entrevista previamente realizada, para así obtener una visión más específica por parte de cada tipo de usuario que pueda pertenecer a alguno de los grupos a los cuales la aplicación va dirigida.</p>
Para aspirantes
<ul style="list-style-type: none"> - ¿Le gustaría encontrar información sobre el programa de su interés de manera detallada? ¿Por qué? - ¿Qué tipo de información le gustaría ver con respecto a cada programa? - ¿Le parece bien que la aplicación cuente con ciertos enlaces a las páginas principales de la universidad, para casos como la información financiera, información de inscripciones, etc., esto debido a que no se pretende repetir información existente y que dicha información siempre esté actualizada?
Para estudiantes
<ul style="list-style-type: none"> - ¿Le gustaría encontrar información sobre el programa al que pertenece de manera detallada? ¿Por qué?

<ul style="list-style-type: none"> - ¿Qué tipo de información le gustaría ver con respecto a su programa? - ¿Qué opina respecto a la publicación de novedades y eventos relacionados a la facultad y los programas? - ¿Qué opina con respecto a la opción de postular sus proyectos y trabajos de grado para que sean mostrados como destacados para los estudiantes y aspirantes que accedan a la aplicación?
Para profesores
<ul style="list-style-type: none"> - ¿Qué opina de que su información académica sea mostrada en la aplicación? - ¿Qué información le parece pertinente que debería de ser mostrada? - ¿Qué opina de la posibilidad de editar los contenidos programáticos de las asignaturas a su disposición (tener en cuenta que dichos cambios deben ser aprobados en primer lugar por el respectivo director de programa)? - ¿Qué opina de la funcionalidad de poder realizar post y publicaciones, relacionados con eventos y/o cambios, en distintos sectores y aspectos de los programas y/o asignaturas? - Con respecto a la posibilidad de sobresaltar trabajos destacados de los estudiantes, para subirlos a la aplicación y que toda la comunidad pueda acceder a ellos, ¿qué opinión tiene al respecto?
Para personal administrativo
<ul style="list-style-type: none"> - ¿Qué información le parece pertinente como contenido de la aplicación? - ¿Qué opina al respecto de la realización de posts y publicaciones relacionadas con eventos y/o cambios en los programas y asignaturas? - ¿Qué opina con respecto a autorizar y/o administrar una sección de “trabajos destacados”, de la cual hacen parte los trabajos y proyectos realizados por estudiantes, que son destacados por parte de sus profesores? - ¿Qué opina al respecto de una funcionalidad que permita a los usuarios acceder a formatos que pueden necesitar por medio de la aplicación?

Cuadro 4.1: Formato entrevistas

Con el formato de entrevista definido, el siguiente paso consistió en entrevistar personas que cumplieran el perfil de los posibles usuarios que tendría la aplicación web, perfiles los cuales fueron definidos anteriormente en este documento.

A continuación, se muestran los resultados de las entrevistas con cada una de las personas seleccionadas (se muestra agrupado dependiendo del perfil de usuario al que las personas pertenezca):

4.2.1. Aspirantes

Aspirante 1:

Aspirante: Juan Fernando Riascos Sinisterra
Juan Fernando es un estudiante de décimo semestre de ingeniería industrial en la Pontificia Universidad Javeriana Cali. Gracias a su experiencia en la institución, ve con buenos ojos realizar alguno de los posgrados afines a su carrera que ofrece la universidad.
Preguntas y respuestas
- ¿Le gustaría encontrar información sobre el programa de su interés de manera detallada? ¿Por qué? R/ Si, por el hecho de que esto motiva ya que muestra temáticas que se van a tratar, le muestra a que se va a enfrentar y qué aportes le puede hacer para la vida personal y profesional.
- ¿Qué tipo de información le gustaría ver con respecto a cada programa? R/ Contenido de las asignaturas, perfiles de los profesores de cada asignatura, para hacerse una noción sobre ellos.
- ¿Le parece bien que la aplicación cuente con ciertos enlaces a las páginas principales de la universidad, para casos como la información financiera, información de inscripciones, etc., esto debido a que no se pretende repetir información existente y que dicha información siempre esté actualizada? R/ Buena iniciativa, debido a que la aplicación sería un complemento que gradualmente se vuelva "oficial" y debido a que recoge todos los enlaces, facilita más la experiencia de navegación.
Recomendaciones
Las funcionalidades son suficientes, ya que como aspirante, principalmente se requiere solo la info.

Cuadro 4.2: Aspirante 1

Aspirante 2:

Aspirante: Yan Carlos Certuche Grueso
Yan Carlos es un estudiante de décimo semestre de ingeniería de sistemas y computación en la Pontificia Universidad Javeriana Cali. En un futuro, después de terminar su carrera, le gustaría cursar uno de los posgrados ofrecidos en la PUJ Cali, afines a su pregrado.
Preguntas y respuestas
- ¿Le gustaría encontrar información sobre el programa de su interés de manera detallada? ¿Por qué? R/ Si, porque uno quiere encontrar la info lo más rápido posible, que esté a la mano. Actualmente no se da la información de manera concreta ni detallada. La información bien detallada puede influir en las decisiones de los interesados.
- ¿Qué tipo de información le gustaría ver con respecto a cada programa?

<p>R/Las secciones de financiación y el contenido, es lo que las personas miran más para poder decirse a cursar el programa. Horarios de clases.</p> <p>- ¿Le parece bien que la aplicación cuente con ciertos enlaces a las páginas principales de la universidad, para casos como la información financiera, información de inscripciones, etc., esto debido a que no se pretende repetir información existente y que dicha información siempre esté actualizada?</p> <p>R/ Es una solución, pero le gustaría encontrar la información directamente por miedo a que los enlaces sean de información muy general. Que tenga sección de contacto para temas financieros. Solo si el enlace lo lleva un lugar bastante específico.</p>
Recomendaciones
Filtrado por departamentos, filtrado nivel de programa (maestría, especialización), enlaces a gestión financiera, estilo similar a la página oficial, información de contacto.

Cuadro 4.3: Aspirante 2

4.2.2. Estudiantes

Estudiante 1:

Estudiante: Diana Muñoz
Diana es estudiante de uno de los programas de posgrado de la facultad de Ingeniería y Ciencias, más específicamente, del programa de Maestría en Ingeniería de Software.
Preguntas y respuestas
<p>¿Le gustaría encontrar información sobre el programa al que pertenece de manera detallada? ¿Por qué?</p> <p>R/ Si, me gustaría bastante.</p> <p>¿Qué tipo de información le gustaría ver con respecto a su programa?</p> <p>R/ Todas las materias y sus módulos, información de cada materia, tecnologías necesarias para distintos cursos. Sería súper. También organización de horarios y horarios disponibles de cursos.</p> <p>¿Qué opina respecto a la publicación de novedades y eventos relacionados a la facultad y los programas?</p> <p>R/ Si, no solo estudiantes resaltados, también información relacionada a la biblioteca, sobre recursos nuevos, novedades académicas y novedades laborales.</p> <p>¿Qué opina con respecto a la opción de postular sus proyectos y trabajos de grado para que sean mostrados como destacados para los estudiantes y aspirantes que accedan a la aplicación?</p> <p>R/ Muy necesario, ejemplos para realizar tesis y todo tipo de trabajos. No solo los resaltados.</p>
Recomendaciones
Marketing de empresas, ayuda de búsquedas de trabajo

Cuadro 4.4: Estudiante 1

4.2.3. Profesores

Profesora 1:

Profesora: María Constanza Pabón
María Constanza es una docente perteneciente a la Facultad de Ingeniería y Ciencias, principalmente asociada al Departamento de Electrónica y Ciencias de la Computación. Además es Directora de los programas de Especialización y Maestría en Ingeniería de Software.
Preguntas y respuestas
¿Qué opina de que su información académica sea mostrada en la aplicación? R/ Si, me parece que debe haber mucho detalle sobre los programas, pero no exactamente el syllabus.
¿Qué información le parece pertinente que debería de ser mostrada? R/ Buena descripción, resumen de los cursos y cada uno de sus ítems.
¿Qué opina de la posibilidad de editar los contenidos programáticos de las asignaturas a su disposición (tener el cuenta que dichos cambios deben ser aprobados en primer lugar por el respectivo director de programa)? R/ Con una aprobación, permitiría mantener actualizados los syllabus.
¿Qué opina de la funcionalidad de poder realizar post y publicaciones, relacionados con eventos y/o cambios, en distintos sectores y aspectos de los programas y/o asignaturas? R/ Sería interesante, organización por temas pero no mucha para no abrumar a las personas con contenidos. Que se pueda filtrar por intereses.
Con respecto a la posibilidad de sobresaltar trabajos destacados de los estudiantes, para subirlos a la aplicación y que toda la comunidad pueda acceder a ellos, ¿qué opinión tiene al respecto? R/ Muy buena, no hay repositorio. Lo manejan las secretarías de posgrado internamente. No hay algo disponible al público.
Recomendaciones
Perfiles (autenticación). Estados de autorizaciones para cambio de contenidos.

Cuadro 4.5: Profesora 1

4.2.4. Personal Administrativo

Personal Administrativo 1:

Miembro personal administrativo: Johana Cárdenas
Johana es la secretaria del departamento de posgrados de la facultad de ingeniería y ciencias de la Pontificia Universidad Javeriana Cali. Debido a su labor, está muy en contacto con muchos de los procesos que se realizan en la facultad.
Preguntas y respuestas
- ¿Qué información le parece pertinente como contenido de la aplicación? R/ Es nueva en el cargo, importante el tema financiero de los estudiantes, horarios, documentación formatos.
- ¿Qué opina al respecto de la realización de posts y publicaciones relacionadas con eventos y/o cambios en los programas y asignaturas? R/ Muy importante.
- ¿Qué opina con respecto a autorizar y/o administrar una sección de “trabajos destacados”, de la cual hacen parte los trabajos y proyectos realizados por estudiantes, que son destacados por parte de sus profesores? R/ Vital, porque es un estímulo para los estudiantes, ya que motiva el acceso.
- ¿Qué opina al respecto de una funcionalidad que permita a los usuarios acceder a formatos que pueden necesitar por medio de la aplicación? R/ Por la situación se busca esto, ir a la universidad es difícil por distancia y tiempo. Es muy importante dar fácil acceso ya que hoy en día es difícil en la plataforma actual.
Recomendaciones
Sin recomendaciones adicionales.

Cuadro 4.6: Miembro personal administrativo 1

Personal Administrativo 2:

Miembro personal administrativo: Carolina Velasco
Carolina es la asistente de la oficina de posgrados de la facultad de ingeniería y ciencias de la Pontificia Universidad Javeriana Cali. Debido a su labor, está muy en contacto con muchos de los procesos que se realizan en la facultad.
Preguntas y respuestas
- ¿Qué información le parece pertinente como contenido de la aplicación? R/ Para aspirantes y estudiantes: plan de estudios, información de profesores, valores y precios de certificados y supletorios, facilidad de búsqueda de la información, derechos de grado, acceso a trabajos, grupos de investigación, documentos para trabajo de grado, instrucciones para subir el trabajo de grado, todos los formatos que puedan necesitar, información salones de clase y horarios. Para colaboradores: accesibilidad a formatos, información de procedimientos actualizada.
- ¿Qué opina al respecto de la realización de posts y publicaciones relacionadas con eventos y/o cambios en los programas y asignaturas? R/ Muy bueno, publicación de sustentaciones, de congresos, seminarios.

- ¿Qué opina con respecto a autorizar y/o administrar una sección de “trabajos destacados”, de la cual hacen parte los trabajos y proyectos realizados por estudiantes, que son destacados por parte de sus profesores?

R/ Importante, para tener acceso más rápido.

- ¿Qué opina al respecto de una funcionalidad que permita a los usuarios acceder a formatos que pueden necesitar por medio de la aplicación?

R/ Importante, ya que evitaría confusiones con respecto a versiones de formatos, fácil acceso para mejorar tiempos.

Recomendaciones

Sin recomendaciones adicionales.

Cuadro 4.7: Miembro personal administrativo 2

De acuerdo a las respuestas brindadas por los entrevistados, se ve que están bastante de acuerdo con la visión planteada por el stakeholder principal, pero sin dejar de lado recomendaciones propias en algunos casos, las cuales fueron evaluadas con Juan Carlos Martínez para verificar su validez y de esta forma, avanzar hacia el siguiente paso del proceso de elicitación: la elicitación de las interacciones.

4.3. Interacciones

Para el proceso de elicitación y validación de las interacciones del sistema, se diseñó un prototipo de cómo se podría ver la aplicación. Este prototipo se diseñó mediante la herramienta web **proto.io**, de la cual se habla un poco más posteriormente en este documento. Este elemento tenía como objetivo servir de base para lo que sería la aplicación al final, visualmente hablando.

La idea de este prototipo era mostrarle a los entrevistados de la etapa anterior, la forma en la que se plasmaron los requisitos establecidos con el stakeholder, y las recomendaciones dadas por ellos mismos. En la figura 4.1, se observa una de las pantallas.

Para ver el resto de pantallas de este prototipo, por favor dirigirse a la sección de anexos del prototipo 2, del apéndice A.

Ahora, el proceso de elicitación de las interacciones del sistema consistió en entrevistar nuevamente a las personas entrevistadas en la etapa de validación de las funcionalidades. Esto con el objetivo de mostrarles la idea de como se plasmarían las funcionalidades en el apartado gráfico, además de mostrarles cuáles de sus opiniones y recomendaciones habían sido tenidas en cuenta.

El proceso que se siguió en estas entrevistas consistió en mostrarle a cada una de las personas, dependiendo el tipo de usuario, el flujo de la aplicación y qué interacciones deberían realizar.

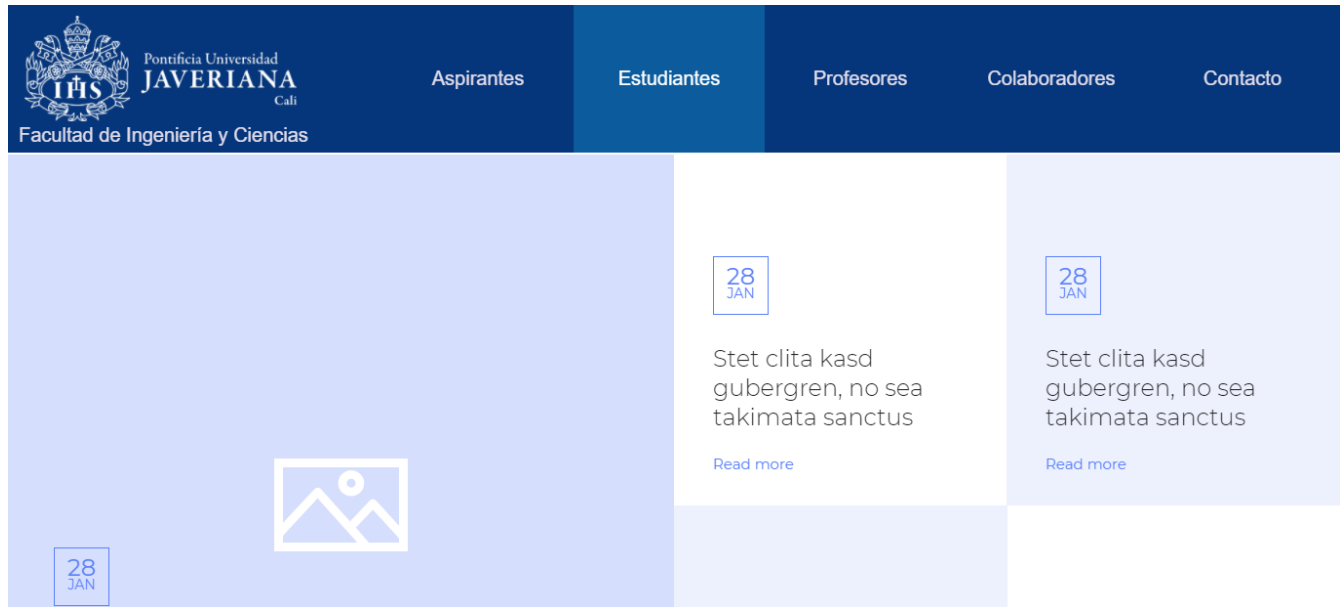


Figura 4.1: Primer prototipo de la aplicación

Paso seguido, se volvió al inicio del prototipo y se le preguntaba a los entrevistados, pantalla por pantalla, cuál era su opinión al respecto de dicha pantalla, para de esta manera, poder recoger un feedback bastante detallado de cada una de las pantallas del prototipo. A continuación se muestran los comentarios de cada uno de los entrevistados:

4.3.1. Aspirantes

Aspirante 1: Juan Fernando Riascos

- **Inicio:** interfaz bien, concisa, es inclusiva con todas las edades.
- **Aspirantes:** está bien para empezar, se puede aprovechar mejor el espacio (imágenes o recuadros).
- **Especializaciones:** está ok, no imágenes porque satura, le da un nivel de escalabilidad bueno.
- **Especialización en Ingeniería de Software:** Agregar indicación de de deslizar.
- **Asignaturas:** lo mismo de especializaciones.
- **Asignaturas específicas:** instrucciones de clic para ver la información.

Aspirante 2: Yan Carlos Certuche

- **Inicio:** ok por el feed de noticias. Interesante ofertas laborales o investigación, becas.
- **Aspirantes:** recuadro horizontal sería mejor visualmente.
- **Especializaciones:** si la lista es extensa, agregar un buscador, de resto normal.
- **Especialización en Ingeniería de Software:** botón para saber que se desliza.
- **Asignaturas:** ok, puede ponerse buscador.
- **Asignaturas específicas:** Ok.

4.3.2. Estudiantes**Estudiante 1: Diana Muñoz**

- **Inicio:** está bien que sea general, visualmente me gusta.
- **Estudiantes:** pueden ser íconos para cada elemento.
- **Maestrías:** resumen desplegable de cada uno de los programas (créditos, acreditación, vínculos) o pueden ser iconos.
- **Maestría en Ingeniería de Software:** chevere el carrusel.
- **Asignaturas:** : le parece bien la lista, antes de dar clic, una nubecita con información de las materias.
- **Asignaturas específicas:** Está chévere. En uno de los cuadros, para poner link de clase o moodle, en caso de que sea virtual.

4.3.3. Profesores**Profesora 1: María Constanza Pabón**

- **Inicio:** Tamaño de la fuente un poco más grande. Es agradable.
- **Profesores:** está bien.
- **Inicio profesores:** lista de eventos creados y cursos o iconos con descripción. Llenar un poco más la pantalla.

- **Cursos:** lista de cursos en pantalla anterior, para disminuir los clics..
- **Curso específico:** : que se muestre más información para que no se vea tan vacío.
- **Recomendaciones:** extraer información de los sistemas de la U para funciones como horarios e información de los cursos.

4.3.4. Personal administrativo

Miembro de personal administrativo 1: Carolina Velasco

- **Inicio:** Bien, pero poner letra más visible.
- **Colaboradores:** está bien, me gusta la simpleza.
- **Inicio colaboradores:** se pueden poner íconos más pequeños o imágenes.
- **Crear post:** está bien, le gusta la idea de limitar el acceso (validación de perfiles).

Como se puede observar, en general a todos los entrevistados les agrada la interfaz presentada en el prototipo, pero de igual manera, algunos dieron recomendaciones importantes, tanto en el aspecto visual, como para el aspecto funcional y/o de contenido. Esta información se validó con el stakeholder principal, el cual estuvo de acuerdo con muchos de los consejos brindados por los entrevistados.

Ahora, con ambos procesos de elicitación y validación de funcionalidades e interacciones, el paso siguiente consistió en, de las entrevistas realizadas, sacar nuevas funcionalidades e interacciones para discutir las y negociarlas con el stakeholder principal.

4.4. Funcionalidades e interacciones adicionales

Después de realizar los procesos de las entrevistas, se obtuvieron datos y recomendaciones que pueden ser tomadas como nuevas funcionalidades y/o interacciones para el desarrollo de la aplicación. A continuación, de cada uno de los dos procesos de entrevistas, se listan aquellos comentarios que clasificaron como nuevas funcionalidades o interacciones:

4.4.1. Funcionalidades

- El sistema debe permitir realizar filtrados en sus distintas secciones de programas y asignaturas.
- El sistema debe mostrar los perfiles de los profesores de cada asignatura.

- El sistema debe mostrar publicaciones, relacionadas con novedades académicas u oportunidades laborales
- El sistema debe realizar un proceso de autenticación en el caso de los usuarios tipo profesor y personal administrativo.
- El sistema debe permitir a los usuarios de tipo personal administrativo, gestionar formatos de documentos para trámites.
- El sistema debe extraer información de los sistemas de la universidad. Información como horarios de clase, cursos, etc.
- El sistema debe tener información de contacto.

4.4.2. Interacciones

- El sistema debe tener un tamaño de fuente legible en todas sus pantallas
- El sistema debe tener instrucciones en algunas de sus secciones, para facilitar la navegación en el mismo.
- El sistema debe contar con buscadores, que faciliten la navegación en secciones que tengan listas muy extensas.
- El sistema debe desplegar información de cada programa o asignatura cuando el cursor pare sobre él.
- El sistema debe tener un aspecto similar a las páginas oficiales de la PUJ Cali.

4.4.3. Validación de funcionalidades e interacciones adicionales

En conjunto con el stakeholder principal, se validaron las nuevas funcionalidades e interacciones que surgieron de los procesos de entrevistas, con el objetivo de verificar si realmente tenían cabida en el proyecto,

Dicho proceso dejó resultados bastante positivos, ya que se acordó con él que, la mayoría de las nuevas funcionalidades e interacciones que habían surgido de los procesos de entrevista, fácilmente cabían en el espectro del proyecto.

4.5. Segundo prototipo

Para continuar con el proceso de levantamiento de información, se realizó un segundo prototipo, basado en los apuntes y ajustes sugeridos por los entrevistados de la etapa anterior. Al ser una etapa más avanzada del proyecto, dicho prototipo se realizó de una manera más completa

y robusta, esta vez, mediante la utilización de un framework de desarrollo frontend llamado ReactJS (para la parte gráfica) y mediante NodeJS, el cual es un entorno de ejecución del lenguaje de programación JavaScript, para la realización del backend (parte lógica de la aplicación), además del uso del motor de bases de datos no relacional MongoDB (En un capítulo posterior se habla un poco más en detalle de las herramientas aquí mencionadas). A continuación, se muestra la figura 4.2, la cual corresponde al segundo prototipo de la aplicación (el resto de capturas se encuentra en la sección de anexos del prototipo 2, del apéndice B), el cual se desarrolló teniendo en cuenta los requisitos que más aportan recolectados en las entrevistas realizadas anteriormente:



Figura 4.2: Segundo prototipo de la aplicación

Para el desarrollo de este prototipo se tuvieron en cuenta recomendaciones como aumentar el tamaño de la fuente, convertir lista en tarjetas y/o imágenes y eliminar algunas pantallas que sobraban o cuyo contenido podía ser incluido fácilmente en una pantalla existente.

Después del desarrollo de este prototipo, el paso a seguir consistió en entrevistar a las mismas personas que se han entrevistado durante todo el proceso, con el objetivo de darles a conocer esta versión más robusta de la aplicación, además de hacerles saber cuáles de sus recomendaciones fueron tenidas en cuenta a la hora de realizar el diseño y desarrollo del mismo.

4.5.1. Entrevistas

Para este proceso se utilizó la misma técnica que en las entrevistas anteriores, el cual consistía en enseñar el flujo que utilizaría cada tipo de usuario dependiendo del entrevistado, para que de esta forma, pudiera dar feedback y recomendaciones de cada una de las pantallas incluidas en dicho flujo. A continuación se muestran los resultados de dichas entrevistas:

4.5.1.1. Aspirantes

Aspirante 1: Juan Fernando Riascos

- **Inicio:** está bien debido a que es muy similar a las páginas oficiales de la universidad.
- **Aspirantes:** está bien y se nota que se siguieron las sugerencias de la entrevista anterior.
- **Especializaciones:** está bien, ya que se mantiene una estructura pensando en que en el futuro se pueden agregar más programas
- **Especialización en Ingeniería de Software:** está bien, ya que cuenta con señalización. Se debería agregar el hipervínculo en la imagen también.
- **Asignaturas:** está bien, porque da pre para poder agregar más en el futuro. Se podría agregar una pestaña que muestre el pensum del programa.
- **Asignaturas específicas:** las tarjetas podrían ser verticales para lograr una mejor distribución del espacio.

Aspirante 2: Yan Carlos Certuche

- **Inicio:** es bueno porque es similar a las redes sociales que utilizan las personas cotidianamente se podrían rellenar los espacios blancos con widgets.
- **Aspirantes:** está chévere, pero se le podría añadir un mejor estilo al título, además, las tarjetas están muy pegadas a los bordes.
- **Especializaciones:** se le puede mejorar el estilo al título, el resaltado de los elementos de la lista podría ser más corto y se podría agregar más información a la pantalla.
- **Especialización en Ingeniería de Software:** las imágenes son muy grandes y están muy pegadas a los bordes.
- **Asignaturas:** exactamente lo mismo que la pantalla de especializaciones. Además, se podría agregar un popo que muestre la información básica de cada curso.
- **Asignaturas específicas:** las tarjetas están muy pegadas a los bordes.

4.5.1.2. Estudiantes

Estudiante 1: Diana Muñoz

- **Inicio:** me gustaría que la barra fuera más grandes, incluyendo el icono y los elementos de la misma. Se deberían rellenar los bordes blancos.
- **Estudiantes:** me gusta pero se deberían rellenar los espacios blancos.
- **Maestrías:** se podría convertir la lista en un carrusel deslizable además de agregar pop ups con la información de cada programa.
- **Maestría en Ingeniería de Software:** me gusta.
- **Asignaturas:** pop up con la información de cada curso.
- **Asignaturas específicas:** Ponerle un estilo más llamativo a las tarjetas.

4.5.1.3. Profesores

Profesora 1: María Constanza Pabón

- **Inicio:** Organizar las noticias en orden cronológico. Disminuir tamaño de las tarjetas de las noticias y de sus imágenes. Limitar la cantidad de noticias que se muestran y agregar un botón para acceder a noticias más antiguas.
- **Profesores:** está bien.
- **Inicio profesores:** ajustar el tamaño de las tarjetas de acuerdo a la extensión de su contenido.
- **Curso específico:** agregar botones para editar noticias y trabajos destacados agregados previamente.
- **Agregar noticia:** mostrar en la parte inferior las noticias agregadas recientemente, para de esa forma, confirmar que se realizó correctamente.

4.5.1.4. Personal administrativo

Miembro de personal administrativo 1: Carolina Velasco

- **Inicio:** está muy vacía, se podrían rellenar los espacios en blanco con alguna información.
- **Colaboradores:** está bien.
- **Inicio colaboradores:** está bien, pero debería tener un poco más de contenido para que no se vea tan vacía.

- **Crear post:** está bien.

Después de realizar todo el proceso de las entrevistas, se obtuvieron más funcionalidades e interacciones que aportan en distintos aspectos de la aplicación. En secciones como las de aspirantes, estudiantes o profesores.

Adicionalmente, se pudo evidenciar que las personas están muy a gusto con el estilo visual de la página y con sus funcionalidades, ya que en la mayoría de los casos, las personas simplemente realizaban pequeñas recomendaciones de tipo estético o ajustes pequeños y/o adicionales a las funcionalidades.

A continuación se sintetizan los resultados de las entrevistas con el objetivo de presentar de manera formal los requisitos obtenidos.

4.5.2. Más funcionalidades

De las entrevistas realizadas en base al prototipo 2, se obtuvieron las siguientes funcionalidades adicionales para la aplicación:

- El sistema debe de mostrar el pensum de cada uno de los programas.
- El sistema debe permitir a los usuarios de tipo profesor subir una foto en la sección de editar información personal.
- El sistema debe mostrar las noticias en las secciones del inicio y de los cursos en orden cronológico (de la más reciente a la mas antigua)
- El sistema debe limitar la cantidad de noticias que muestre en el inicio y en cada curso.
- El sistema debe de tener una función que permita tener acceso a todas las noticias, incluyendo las más antiguas, mediante un botón.
- El sistema debe tener una función que permita editar noticias que fueron creadas anteriormente.
- El sistema debe tener una función que permita editar trabajos destacados agregados anteriormente.

4.5.3. Más interacciones

De las entrevistas realizadas con base al segundo prototipo, también se obtuvieron nuevas interacciones, las cuales son las siguientes:

- La pantalla de inicio del sistema debe de tener widgets y/o herramientas que rellenen los espacios blancos de los bordes.

- Las tarjetas de varias pantallas del sistema deben estar posicionadas un poco más alejadas de los bordes de la pantalla.
- El título de algunas pantallas del sistema debe ser más llamativo.
- Se debe ajustar el tamaño de algunas de las imágenes presentadas en algunas pantallas del sistema.
- El sistema deberá mostrar una nube o pop up con la información básica de cada programa en la sección de los programas de cada nivel de posgrado.
- Las imágenes del sistema deberán estar asociadas a los mismos vínculos que sus títulos y/o descripciones.
- La barra de navegación del sistema debe ser un poco más grande.
- El sistema deberá mostrar una nube o pop up con la información básica de cada asignatura dentro de la pantalla de cada programa.
- Las tarjetas de las distintas pantallas del sistema deben ser un poco más estilizadas.
- Se debe ajustar el tamaño de las noticias de la pantalla de inicio del sistema.
- Se debe ajustar el tamaño de las tarjetas de la pantalla de cada curso del sistema, de acuerdo a la cantidad de información que contengan.

Como se puede observar en los dos casos anteriores, gracias que el prototipo se realizó de una manera más compleja y con herramientas de desarrollo de software, permitió recolectar una buena cantidad de funcionalidades e interacciones mucho más específicas y cercanas a la realidad, lo cual es un gran aporte para que la aplicación sea cada vez más completa.

Herramientas utilizadas para el prototipado

Para la realización de los prototipos del proyecto se tuvieron en cuenta dos enfoques totalmente diferentes: en primer lugar, una aplicación web que permite realizar prototipos de todo tipos de aplicaciones de una manera muy sencilla, pero sin agregar funcionalidad, y por otro lado, el uso de frameworks de lenguajes de programación para desarrollar una etapa temprana de la aplicación que funcionaba también como un prototipo más sofisticado y robusto que el primero.

5.1. Primer prototipo

Como se menciona arriba, para la elaboración del prototipo inicial de la aplicación se utilizó una herramienta web conocida como proto.io [22], la cual cuenta con una gran cantidad de herramientas y funcionalidades que permiten el diseño de interfaces de usuario, con el objetivo de utilizarlas como prototipos para desarrollos posteriores. Suele ser muy utilizada en etapas tempranas de proyectos de desarrollo de software, con el objetivo de adquirir nociones de cómo debe lucir la aplicación a desarrollar en el futuro, y además de qué herramientas usar. A continuación se muestra en la figura 5.1, la pantalla de inicio de proto.io, donde se pueden apreciar algunas de su funcionalidades:

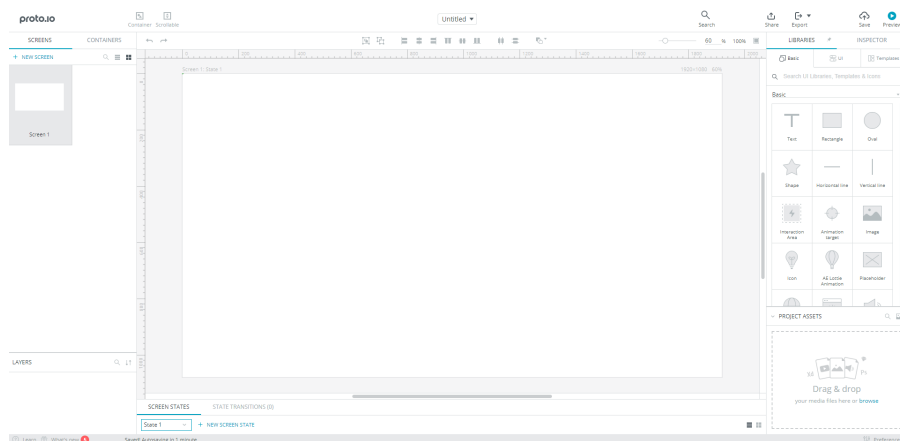


Figura 5.1: Herramienta proto.io

5.2. Segundo prototipo

Para el segundo prototipo, ya se contaba con una buena cantidad de feedback obtenida gracias a las primeras entrevistas realizadas con el primer prototipo. Debido a lo anterior, y como también se querían mostrar funcionalidades del sistema, se decidió que el segundo prototipo fuera desarrollado en un lenguaje de programación, ya que de esta forma además de contar con el aspecto visual de la aplicación también se tendría en cuenta el aspecto funcional y lógico de la aplicación, con el objetivo de que en la siguiente ronda de entrevistas se consiguiera un nivel de feedback más detallado, que es exactamente lo que ocurrió.

Teniendo en cuenta lo dicho anteriormente, como herramientas para el desarrollo del segundo prototipo se seleccionaron 3: React.js (framework para desarrollo del frontend del prototipo), Node.js (framework para el desarrollo del backend del prototipo) y MongoDB (motor de base datos para almacenar la información necesaria para mostrar en el prototipo). La razón por las que se escogieron estas 3 herramientas fue debido a los conocimientos previos y familiaridad que se tenía con las mismas.

5.2.1. React.js

React.js [23] es un framework para desarrollo frontend del lenguaje de programación JavaScript, orientado a componentes, es decir, que cada objeto que se renderiza en pantalla es un componente definido dentro del desarrollo.

Para este proyecto, el enfoque de componentes fue de gran utilidad, ya que brindaba cierto nivel de sencillez para implementar cambios y recomendaciones realizadas por las personas entrevistadas.

5.2.2. Node.js

Node.js [24] es un entorno de ejecución para el lenguaje de programación JavaScript, orientado principalmente al desarrollo de aplicaciones network escalables, es decir, aplicaciones que son desarrolladas con el objetivo de volverse más robustas y completas con el pasar del tiempo.

El enfoque hacia la escalabilidad que brinda Node.js, fue de gran ayuda para este proyecto, ya que a medida que se avanzaba en los proceso de desarrollo, entrevistas y reuniones con el director del proyecto, brindaba la facilidad de ir aumentando el tamaño de la aplicación de una manera sencilla, basado en los comentarios que se iban obteniendo y de esa manera poder agregarle más funcionalidades a algunos de los componentes desarrollados en React.js, y así hacer el prototipo cada vez más funcional

5.2.3. MongoDB

MongoDB [25] es un motor de bases de datos orientado a documentos, es decir, la forma en la que almacena la información es estructuras de tipo JSON, lo que lo hace eficaz para almacenar cualquier tipo de información que sea necesaria. Elimina la necesidad de tener una cantidad innecesaria de tablas al reemplazarlas por colecciones, que son los conjuntos en los cuales se almacenan los documentos, los cuales dan la libertad de no tener que tener la misma estructura.

Dentro del proyecto, fue una muy buena elección como motor de base de datos, ya que con el tipo de información que maneja, se hubieran tenido que crear una buena cantidad de tablas, pero a la final esto se paso a un número reducido de colecciones, con el agregado de la facilidad de manejo de la información que otorgan los JSON.

Análisis comparativo de dos procesos de ingeniería de requisitos

Después de realizar el proceso de ingeniería de requisitos con la metodología seleccionada para el proyecto, Agile UX, el paso a seguir consistió en realizar un análisis comparativo entre dicho proceso y un proceso similar realizado por estudiantes pertenecientes al programa de Maestría en Ingeniería de Software de la Pontificia Universidad Javeriana Cali, más específicamente, estudiantes pertenecientes al curso de Ingeniería de Requisitos. El proceso realizado por ellos estaba relacionado al trabajo final de dicho curso fue realizado teniendo como base los procesos de ingeniería de requisitos relacionados a las metodologías tradicionales de desarrollo.

6.1. Proceso realizado con ingeniería de requisitos de metodologías tradicionales

El proceso de ingeniería de requisitos realizado por los estudiantes de la Maestría en Ingeniería de Software Luis Santiago Osorio Ortiz, Harold William Victoria Sandoval y Cesar Augusto Neira Guzmán [26] se basa en la implementación de un SRS, es decir, un documento de especificación de requisitos de software [27], el cual incluye una serie de recomendaciones para la especificación y elicitación de requisitos para la realización de un proyecto de software, y cuyo principal objetivo es formalizar los acuerdos entre el equipo encargado de la realización del proyecto y el cliente para el cual trabajan.

El SRS tiene una estructura que determina qué se debe hacer en cada etapa del proceso de ingeniería de requisitos. Plantea que se deben definir requisitos funcionales, no funcionales, requisitos de restricción y requisitos de datos. Adicionalmente, incluye la definición de casos de uso, además de la definición de conceptos asociados al proyecto como puede ser el contexto, dominio, características del usuario, restricciones y dependencias.

Teniendo en cuenta lo anterior, una de las principales diferencias del proceso realizada por los estudiantes de maestría y el proceso que se realizó en este proyecto, fue la inclusión de dos tipos de usuarios adicionales por parte de ellos, además que es un proceso mucho más formal debido a todos los conceptos y definiciones que se incluyen inicialmente.

El primer paso realizado por los estudiantes de maestría consistió en la definición de los conceptos, elementos y características relacionados al proyecto que se mencionaron anteriormente. El paso a seguir consistió en iniciar el proceso de definición de requisitos, el cual fue realizado de acuerdo a lo que quería el cliente en ese momento. A continuación, se presentan los requisitos definidos inicialmente por los estudiantes de maestría:

6.1.1. Requisitos

Según el formato SRS manejado por los estudiantes de maestría, debían definirse requisitos funcionales, no funcionales, de restricciones y de datos, teniendo en cuenta la información recolectada en charlas previas con el cliente.

6.1.1.1. Requisitos funcionales

A continuación, se muestran los requisitos funcionales definidos por los estudiantes de maestría:

- RF-01: El sistema debe permitir la administración de usuarios, su activación, inactivación, actualización y asignación de privilegios para la publicación de contenidos según su clasificación. Los tipos de usuarios que deberá manejar el sistema son:
 - RF-01-01: Visualizador: Quienes consultan el contenido público sin ningún tipo de restricción (Aspirantes, público en general).
 - RF-01-02: Alumno: Quienes consultan el contenido público y el contenido protegido (Estudiantes y Egresados de la PUJC), su autenticación se realizará con el sistema de Identidad de la PUJC.
 - RF-01-03: Docente: Quienes pueden crear, editar y eliminar el contenido (Directores, Docentes y Asistentes), su autenticación se realizará con el sistema de Identidad de la PUJC.
 - RF-01-04: Administrador: Quienes Administran la aplicación y los perfiles de usuarios. Su autenticación se realizará con el sistema de Identidad de la PUJC.
- RF-02: El sistema debe permitir a los usuarios realizar una búsqueda rápida de los contenidos publicados a partir de una cadena de texto.
- RF-03: El sistema debe permitir administrar el ciclo de vida de los contenidos. Debe permitir crear el contenido, guardarlo como borrador, gestionarlo (editar) y publicarlo.
- RF-04: El sistema debe permitir la recepción de correos electrónicos desde un formulario de contacto.
- RF-05: El sistema debe permitir la gestión de menús para permitir la organización de los contenidos.

- RF-06: El sistema debe permitir que los contenidos contengan archivos adjuntos de tipo documento ofimático y multimedia. Debe realizar la validación para impedir el cargue de archivos ejecutables.
- RF-07: El sistema debe permitir que los usuarios descarguen los archivos adjuntos.
- RF-08: El sistema debe tener un módulo de Preguntas Frecuentes (FAQ).
- RF-09: El sistema deberá tener un Blog para la publicación de información relevante.
- RF-10: El sistema debe realizar el conteo de todas las consultas realizadas a cada contenido.
- RF-11: El sistema debe permitir la generación de reportes que permitan consultar:
 - RF-11-01: Número de visualizaciones a cada contenido.
 - RF-11-02: Número de visualizaciones a cada etiqueta.
 - RF-11-03: Número de usuarios suscritos y/o cancelados a cada contenido.
 - RF-11-04: Número de usuarios suscritos y/o cancelados a cada etiqueta.
 - RF-11-05: Número de usuarios dados de alta y/o baja en un periodo de tiempo.
 - RF-11-06: Número de usuarios que han actualizado datos en un periodo de tiempo.
 - RF-11-07: Número de visualizaciones a cada entidad.
- RF-12: El sistema deberá tener un foro para la publicación de información relevante.
- RF-13: El sistema debe contar con una opción de chat. Esta opción iniciará un chatbot automático donde se presentarán las 10 preguntas más frecuentes previamente definidas. En el momento en que el usuario pregunte sobre información que no se encuentre en el inventario inicial de preguntas frecuentes, el sistema deberá responder con un enlace al sistema de mensajería en línea, el cual debe administrar y encolar los mensajes cuando no exista un usuario para responder de manera inmediata.
- RF-14: El sistema debe contar con un proceso automático que verifique la correcta respuesta de los enlaces (links) que estén publicados en los contenidos del sistema.
 - RF-14-01: El sistema debe permitir que la ejecución del proceso se pueda programar de manera periódica.
 - RF-14-02: El sistema debe enviar una notificación con el resumen de la prueba a una lista de correos configurables cuando la validación del enlace genere un error.
- RF-15: El sistema debe permitir el manejo de diferentes idiomas en sus contenidos.
- RF-16: El sistema debe permitir la creación, actualización, inactivación y eliminación de diferentes tipos de entidades que permitan definir la organización y los contenidos:
 - Facultad: Entidad de máximo orden jerárquico que contiene uno o más departamentos.

- Departamento: Entidad en la que se descomponen las facultades y que contiene uno o más programas.
 - Programa: Entidad en la que se descomponen los departamentos y que contiene uno o más semestres. También conocido como carrera.
 - Semestre: Entidad que representa la temporalidad y duración de los programas. Contienen una o más materias.
 - Materia: Entidad que contiene el contenido académico de cada semestre y programa.
 - Persona: Entidad que representa cualquier tercero relacionado con ninguna o más entidades.
 - Docente: Entidad derivada de Persona que representa un usuario con capacidad de gestionar contenidos. Puede estar asociado a una o más materias.
 - Estudiante: Entidad derivada de Persona que representa un usuario que consume contenidos.
 - Empleado: Entidad derivada de Persona que representa un usuario con capacidad de gestionar contenidos. Puede estar asociado a uno o más Programas, departamentos, facultades.
 - Grupo: Entidad que representa la unión de una o varias personas y materias. Está asociado a uno o más programas, departamentos y facultades.
 - Syllabus: Entidad que representa la definición específica de cada Materia.
 - Eventos: Entidad que representa una actividad realizada por una o más materias, programas, departamentos y facultades. Pueden inscribirse a el uno o más personas, docentes, estudiantes, empleados y grupos.
- RF-17: El sistema debe permitir gestionar (crear, editar, borrar) entidades (definidas en el RF-16).
 - RF-18: El sistema debe permitir clasificar los contenidos con etiquetas (hashtags), categorías y subcategorías para clasificar los contenidos. Un contenido puede relacionar más de una etiqueta.
 - RF-18-01: El sistema debe permitir configurar algunas categorías, subcategorías y etiquetas para solicitar revisión y aprobación de un usuario con mayores privilegios.
 - RF-19: El sistema debe permitir el alta, baja y la actualización de datos de los usuarios tipo Visualizador. Para este tipo de usuario se deben persistir el número de identificación (que será el ID de usuario), nombres y apellidos, correo electrónico, teléfono móvil y una contraseña.
 - RF-20: El proceso de alta debe tener dos opciones: Una para el alta de los usuarios tipos Visualizador quienes registran los datos y confirman la suscripción a través de un enlace que se debe enviar al correo electrónico registrado. Y la segunda, para los usuarios que

tienen usuario y clave en la PUJC, quienes serán redirigidos al sistema de Identidad de la PUJC.

- RF-21: El sistema debe contar con una sección de publicaciones que permanezca siempre visible en todos los contenidos del portal. Adicionalmente debe permitir clasificar dichas publicaciones con una etiqueta que permita filtrar las publicaciones dependiendo de la sección en la que se encuentre el usuario.
 - RF-21-01: El sistema debe permitir la suscripción y/o cancelación a cada tipo de contenido y/o etiqueta a los usuarios que estén registrados en el sistema. Cuando se requiera la suscripción de un usuario Visualizador y el usuario no esté dado de alta, se debe obligar al usuario a darse de alta.
 - RF-21-02: El sistema debe notificar a los usuarios suscritos a cada etiqueta o tipo de contenido cuando se publique un nuevo contenido y enviar el link del contenido a los correos electrónicos registrados

Después de definir los requisitos funcionales, realizaron el diagrama de casos de uso, el cual se puede observar en los anexos.

6.1.1.2. Requisitos no funcionales

A continuación, se muestran los requisitos no funcionales definidos por los estudiantes de maestría:

- RNF-01: El sistema debe cumplir con las siguientes normas de usabilidad propuestas por Jakob Nielsen: relación entre el sistema y el mundo real, control y libertad del usuario, consistencia y estándares, estética y diseño minimalista y ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores.
- RNF-02: El sistema debe poseer un diseño “Responsive” para garantizar la adecuada visualización en los siguientes dispositivos: PC’s, Tablets y Smartphones.
- RNF-03: El sistema no deberá implementar ninguna funcionalidad, procedimiento o proceso que ejecute un sistema existente en la PUJC.
- RNF-04: El sistema debe ser capaz de operar adecuadamente con hasta 500 usuarios concurrentes.
- RNF-05: Los contenidos insertados, borrados y/o modificados en la base de datos deben ser actualizados para todos los usuarios que acceden a la zona pública en menos de 2 segundos.
- RNF-06: El sistema debe desarrollarse utilizando patrones, recomendaciones y buenas prácticas definidas por la Universidad.

- RNF-07: El sistema debe operar correctamente en los siguientes exploradores y versión:
 - Google Chrome versión 77.X
 - Microsoft Edge versión 41.X
 - Safari versión 11.X
 - Opera versión 63.X
 - Firefox versión 70.X
 - Internet Explorer versión 11.X
- RNF-08: El sistema debe cargar cada página de navegación en menos de 2 segundos.
- RNF-09: El sistema debe ser implementado en un CMS que permita crear, administrar y gestionar contenidos multimedia.
- RNF-10: El sistema debe manejar fuentes de alfabeto en español e inglés.
- RNF-11: El sistema debe cumplir con la ley de tratamiento en condiciones de igualdad para personas con discapacidad.
- RNF-13: El sistema no debe requerir datos personales de los usuarios ni tener procesos de registro de usuarios al sistema de la PUJC.
- RNF-14: El sistema no debe tener opciones de blackboard.
- RNF-15: El sistema debe ser desarrollado siguiendo la guía OWASP.
- RNF-16: El sistema debe permitir la personalización del entorno con tres temas (estilos) diferentes.
- RNF-17: El sistema debe permitir la exportación e importación del backup con todo el contenido para restablecer el servicio en caso de contingencia.

6.1.1.3. Requisitos de tipo restricción

A continuación, se muestran los requisitos de tipo restricción definidos por los estudiantes de la maestría:

- RES-01: El sistema debe usar el método de autenticación que provee la Universidad Javeriana.
- RES-02: El sistema debe seguir todas las condiciones de imagen de acuerdo con la imagen corporativa.
- RES-03: El sistema debe seguir los estándares de desarrollos requeridos por la universidad (CSI - Centro de servicios informáticos).

- RES-04: Presupuesto de \$30.000.000.
- RES-05: Tiempo de entrega del sistema en producción, 01 de julio de 2020.

6.1.2. Prototipo

Después de la definición de los requisitos, realizaron el diseño de un prototipo de la aplicación, el cual se puede ver en la figura 6.1:



Figura 6.1: Prototipo estudiantes de maestría

Luego del diseño del prototipo, realizaron el proceso de especificación de cada uno de los requisitos.

6.1.3. Requisitos de datos

Posterior a la especificación, definieron los requisitos de datos, los cuales se refieren a la información que se almacenará en la base de datos de la aplicación. Definieron la información

como un conjunto de distintas entidades, las cuales son las siguientes:

- **Facultad:** Entidad de máximo orden jerárquico que contiene uno o más departamentos.
- **Departamento:** Entidad en la que se descomponen las facultades y que contiene uno o más programas.
- **Programa:** Entidad en la que se descomponen los departamentos y que contiene uno o más semestres. También conocido como carrera.
- **Semestre:** Entidad que representa la temporalidad y duración de los programas. Contienen una o más materias.
- **Materia:** Entidad que contiene el contenido académico de cada semestre y programa.
- **Persona:** Entidad que representa cualquier tercero relacionado con ninguna o más entidades.
- **Docente:** Entidad derivada de Persona que representa un usuario con capacidad de gestionar contenidos. Puede estar asociado a una o más materias.

6.1.4. Requisitos eliminados

Al haber definido todos los conjuntos de requisitos que plantea el formato del SRS, el paso siguiente que realizaron fue la eliminación de requisitos que no fueran necesarios, proceso avalado por reuniones con el cliente. Dichos requisitos son:

- RF-03
- RF-04
- RF-05
- RF-06
- RF-07

6.1.5. Estrategias y métodos de elicitación

Como paso siguiente, exponen los métodos y técnicas que utilizaron para la elicitación de los requisitos durante todas las etapas del proceso, los cuales, son los siguientes:

- **Entrevistas:** utilizadas principalmente en los contactos con el cliente o stakeholder principal. Consta de preguntas abiertas para capturar las necesidades del cliente.
- **Cuestionarios:** crearon cuestionarios para conocer el comportamiento de posibles usuarios de la aplicación. Constan de preguntas de selección múltiple.

- **Lluvia de ideas:** proceso que involucró a todos los estudiantes del curso de ingeniería de requisitos de la maestría en ingeniería de software de la Pontificia Universidad Javeriana Cali, en la cual daban su opinión acerca de cuáles funcionalidades podría tener la aplicación.
- **Observación:** se realizó un proceso de observación teniendo en cuenta las aplicaciones existentes de la universidad, con el objetivo de extraer nuevos requisitos o validar requisitos existentes.

6.1.6. Priorización, validación y negociación de requisitos

Con todos los requisitos en su poder, los estudiantes de maestría realizaron 3 procesos: se encargaron de priorizar los requisitos, es decir, de asignarle un nivel de importancia a cada uno, con el objetivo de saber cuáles se deben desarrollar primero; luego, se encargaron de realizar un proceso de validación de aquellos requisitos eliminados anteriormente, con la ayuda de una ingeniera invitada al curso; y, finalmente, realizaron un proceso de negociación, en el cual se reunieron con el cliente para realizar modificaciones a algunos requisitos para de esta forma facilitar algunos aspectos del proceso de desarrollo de la aplicación. A continuación se muestran los requisitos que fueron modificados en el proceso de negociación:

- **Comunicación entre oficinas de postgrado (Actualización RF-13):** El sistema debe tener una opción de chat, se iniciará un chat automático donde se presentarán las 10 preguntas más frecuentes, en el momento en que el usuario pregunte sobre una información que no se encuentre en el inventario inicial el sistema deberá responder con un enlace al sistema de mensajería directa, el cual debe administrar y encolar los mensajes.
- **Zona de noticias (Actualización al RQ RF-12):** El sistema debe contar con una sección de publicaciones que permanezca siempre visible en todos los contenidos del portal, adicionalmente debe permitir clasificar dichas publicaciones con una etiqueta que permita filtrar las publicaciones dependiendo de la sección en la que se encuentre el usuario.
El sistema debe tener una opción de suscripción de publicaciones donde permita seleccionar las diferentes etiquetas de las cuales está interesado el usuario.
El sistema debe notificar a los usuarios suscritos al correo electrónico cada que se realice una publicación de acuerdo con los intereses seleccionados en el momento del registro.
- **Chequeo de enlaces (Actualización al RQ RF-14):** El sistema debe contar con un proceso que verifique automáticamente la correcta respuesta de los links correspondientes al contenido principal del portal. El sistema debe permitir que la ejecución del proceso se pueda programar de manera periódica. En caso de recibir un error el sistema debe enviar una notificación con el resumen de la prueba a una lista de correos configurables.

Después de la realización de estos 3 procesos, se da por terminado procedimiento de ingeniería de requisitos realizado por los 3 estudiantes de maestría.

6.2. Proceso de comparación

Después de realizar el resumen del proceso de ingeniería de requisitos implementado por los estudiantes de maestría, y teniendo en cuenta el proceso realizado en capítulos anteriores de este proyecto, se realizó un comparativo entre los resultados de ambos procedimientos, es decir, se compararon la calidad y cantidad de requisitos que se generaron en ambos casos, teniendo en cuenta una serie de características y criterios teniendo como base los establecido en el artículo *Effective Requirements Development—A Comparison of Requirements Elicitation Techniques* [28], en el que Zhang plantea una serie de criterios basándose principalmente en el tipo de técnicas de elicitación que utiliza cada una de las metodologías que se desean comparar.

En el artículo de Zhang, se puede observar que lo que principalmente realiza, es una comparación entre distintas técnicas de elicitación de requisitos, pero, para efectos de este proyecto, lo que se deseaba comparar eran los resultados obtenidos de aplicar dos metodologías de ingeniería de requisitos distintas a un mismo problema. Por lo tanto, dada la naturaleza de los criterios definidos en el artículo, estos se pudieron adaptar fácilmente para también poder realizar la calificación de procesos de ingeniería de requisitos, con el objetivo de hacer una comparación similar a la que realiza la autora.

Teniendo en cuenta lo anterior, se realizaron unos ajustes a algunos de los criterios, y se agregaron algunos nuevos, avalados y/o recomendados por el director del proyecto durante un proceso de validación de estos criterios. A continuación, se muestra cada uno de los criterios que fueron definidos y una pequeña descripción de cada uno:

- **Nivel de abstracción de requisitos: Análisis del problema:** se refiere al nivel de análisis de la situación que ofrece la metodología, teniendo en cuenta su enfoque y técnicas.
- **Nivel de abstracción de requisitos: Descripción del producto:** se refiere al nivel de detalle que alcanza la descripción de los distintos aspectos de un proyecto con el uso de una metodología, teniendo en cuenta las técnicas y procedimientos que esta utiliza.
- **Fuente de los requisitos (stakeholders): Personas:** se refiere a si los requisitos fueron recolectado por medio de personas.
- **Fuente de los requisitos (stakeholders): Otros ambientes:** se refiere a si los requisitos fueron recolectados por medios como aplicaciones similares, documentación, entre otros.
- **Obstáculos de comunicación: Cultura nacional:** se refiere al nivel de facilidad de comunicación teniendo en cuenta la cultura del país.
- **Obstáculos de comunicación: Cultura organizacional:** se refiere al nivel de facilidad de comunicación teniendo en cuenta la cultura de la organización a la que pertenece la persona.

- **Obstáculos de comunicación: Limitaciones cognitivas:** se refiere al nivel de facilidad de comunicación teniendo en cuenta posibles dificultades cognitivas que pueda tener la persona.
- **Obstáculos de comunicación: Geografía:** se refiere al nivel de facilidad de comunicación teniendo en cuenta la ubicación geográfica de la persona.
- **Nivel de veracidad: Dominio existente:** se refiere a la capacidad de la metodología de adaptarse a un dominio existente, es decir, a algo conocido para el equipo de trabajo.
- **Nivel de veracidad: Dominio nuevo:** se refiere a la capacidad de la metodología de adaptarse a un dominio nuevo, es decir, a algo desconocido para el equipo de trabajo.
- **Cantidad de requisitos funcionales:** se refiere a la cantidad de requisitos funcionales o funcionalidades que permite obtener la metodología.
- **Cantidad de requisitos de interacción o interfaz:** se refiere a la cantidad de requisitos de interfaz o interacciones que permite obtener la metodología.
- **Cantidad de requisitos no funcionales:** se refiere a la cantidad de requisitos no funcionales que permite obtener la metodología.
- **Cantidad total de requisitos:** se refiere a la cantidad total de requisitos que permite obtener la metodología.
- **Cantidad de iteraciones:** se refiere a la cantidad de iteraciones realizadas durante el proceso, es decir, la cantidad de veces que se repiten ciertos procesos para refinar el proyecto.
- **Nivel de contacto con stakeholders:** se refiere al nivel de cercanía que se maneja con los stakeholders del proyecto, es decir, qué tan buena es la comunicación entre ambas partes..
- **Nivel de contacto con usuarios:** se refiere al nivel de cercanía que se maneja con los posibles usuarios del proyecto, es decir, qué tan buena es la comunicación entre ambas partes.
- **Cantidad de reuniones con stakeholders:** se refiere a la cantidad de reuniones que se realizaron con los stakeholders para discutir temas de requisitos.
- **Cantidad de reuniones con usuarios:** se refiere a la cantidad de reuniones que se realizaron con los posibles usuarios para discutir temas de requisitos.

6.2.1. Valoración de los criterios

Para la valoración de los criterios se tuvieron en cuenta un conjunto de 3 calificaciones. A continuación, se muestra como se definió la escala de calificaciones:

- 1: No cumple con el criterio

- 3: Cumple medianamente el criterio
- 5: Cumple definitivamente el criterio

Teniendo en cuenta las calificaciones definidas, el paso siguiente consistió en realizar la calificación de cada uno de los criterios relacionándolos a cada metodología, con el objetivo de obtener un puntaje y así se determinó cuál de las dos metodologías era la más adecuada el desarrollo de aplicaciones web como el caso de estudio aquí presentado y proyectos similares.

6.3. Análisis comparativo

A continuación, se muestra en la tabla 6.1 las calificaciones dadas a cada uno de los criterios teniendo en cuenta los resultados de las dos metodologías de ingeniería de requisitos:

Criterios	Agile UX	Metodología tradicional
Nivel de abstracción de requisitos: Análisis del problema	5	5
Nivel de abstracción de requisitos: Descripción del producto	5	3
Fuente de los requisitos (stakeholders): Personas	5	5
Fuente de los requisitos (stakeholders): Otros ambientes	3	3
Obstáculos de comunicación: Cultura nacional	5	5
Obstáculos de comunicación: Cultura organizacional	5	5
Obstáculos de comunicación: Limitaciones cognitivas	1	1
Obstáculos de comunicación: Geografía	5	5
Nivel de veracidad: Dominio existente	5	5
Nivel de veracidad: Dominio nuevo	1	1
Cantidad de requisitos funcionales	5	3
Cantidad de requisitos de interacción o interfaz	5	1
Cantidad de requisitos no funcionales	1	5
Cantidad total de requisitos	5	3
Cantidad de iteraciones	5	1
Nivel de contacto con stakeholders	5	3
Nivel de contacto con usuarios	5	3
Cantidad de reuniones con stakeholders	5	3
Cantidad de reuniones con usuarios	5	3

Cuadro 6.1: Evaluación de criterios

6.3.1. Justificación de las valoraciones

Después de calificar cada uno de los criterios para ambas metodologías, se debían justificar las razones por las cuales se calificó de esa manera, con el objetivo de darle más peso al com-

parativo. Cada una de las justificaciones se validaron con el director del trabajo y se muestran a continuación:

- **Nivel de abstracción de requisitos: Análisis del problema:** en este criterio se le dio la calificación más alta a ambas metodologías, ya que en ambos casos se puede observar un nivel de detalle bastante completo con respecto a contextualizar el problema tratado en el proyecto, aunque en ambos casos con enfoques distintos.
- **Nivel de abstracción de requisitos: Descripción del producto:** en este criterio se le dio una mejor calificación (la más alta) a la metodología Agile UX, debido a que, al ser una metodología enfocada en la experiencia del usuario, permitió dar una vista más detallada del producto y sus características.
- **Fuente de los requisitos (stakeholders): Personas:** este criterio tuvo la máxima calificación para ambas metodologías, ya que, en ambos casos, la fuente principal de información y de requisitos fueron los stakeholders y los usuarios.
- **Fuente de los requisitos (stakeholders): Otros ambientes:** en este criterio se le dio la calificación intermedia a ambos procesos, ya que la única fuente de requisitos diferentes a personas, utilizada en ambos casos, fue la página web de la Pontificia Universidad Javeriana Cali.
- **Obstáculos de comunicación: Cultura nacional:** en este criterio ambos obtuvieron calificación máxima, debido a que ambos procesos funcionan de la misma forma sin importar la cultura practicada por las personas involucradas.
- **Obstáculos de comunicación: Cultura organizacional:** en este criterio ambos obtuvieron calificación máxima, debido a que ambos procesos funcionan de la misma forma sin importar la organización involucrada.
- **Obstáculos de comunicación: Limitaciones cognitivas:** este criterio tuvo la calificación más baja para ambos casos, ya que en ninguna de los dos procesos se tuvo en cuenta un posible encuentro con personas con limitaciones cognitivas.
- **Obstáculos de comunicación: Geografía:** en este criterio ambos obtuvieron calificación máxima, debido a que ambos procesos funcionan de la misma forma ya que las personas involucradas están ubicadas en el mismo sitio geográficamente (Cali) en ambos casos.
- **Nivel de veracidad: Dominio existente:** para este criterio, la calificación fue la más alta en ambos caso, ya que el objetivo de realizar el proceso de ingeniería de requisitos se mueve en un dominio familiar y ya existente, más específicamente, la facultad de ingeniería y ciencias de la Pontificia Universidad Javeriana Cali.
- **Nivel de veracidad: Dominio nuevo:** en este caso, la calificación del criterio fue la más baja para ambos procesos, ya que en ninguno de los dos se explora un nuevo dominio.

- **Cantidad de requisitos funcionales:** en este caso obtuvo mejor calificación Agile UX, ya que a pesar de que la metodología tradicional contaba con 21 requisitos funcionales inicialmente, después de varios procesos, ese número se redujo a 16, que no está muy lejos de los 19 obtenidos mediante Agile UX, pero igualmente, tener más requisitos al final y validados, genera un agregado importante.
- **Cantidad de requisitos de interacción o interfaz:** en este caso, el criterio obtuvo una nota más alta para Agile UX y la más baja para la metodología tradicional, debido a que la primera es una metodología totalmente enfocada a la experiencia del usuario, y la interacción de los stakeholders y usuarios con los prototipos permitió recolectar una buena cantidad de requisitos (16), mientras que, la metodología tradicional se basaba más en requisitos funcionales y no funcionales.
- **Cantidad de requisitos no funcionales:** para este criterio, la metodología tradicional obtuvo la calificación más alta y Agile UX la más baja, esto debido principalmente al enfoque distinto que se manejó con Agile UX, que se enfocaba más en cómo interactuaba el usuario con los prototipos de la aplicación y dejaba de lado algunos procesos tradicionales como la elicitación de requisitos no funcionales, cosa que la metodología tradicional respetaba completamente (17 requisitos).
- **Cantidad total de requisitos:** teniendo en cuenta los requisitos funcionales, no funcionales y de interacción, se obtuvo que el total de requisitos obtenidos mediante Agile UX fue de 35, y por la metodología tradicional fue de 33, por lo tanto Agile UX obtiene la calificación más alta y la metodología tradicional la calificación intermedia, debido a la cercanía entre ambos totales.
- **Cantidad de iteraciones:** el proceso de la metodología tradicional no cuenta con un proceso iterativo, ya que se basa, según lo observado y documentado, en un método de cascada. Agile UX, al tener su base en las metodologías ágiles, cuenta con un proceso iterativo para la obtención y validación de los requisitos, que en el caso de este proyecto se realizó 3 veces, más específicamente, antes y después de las etapas de prototipado. El aspecto iterativo generó un agregado bastante importante, ya que permitió estar en un contacto más directo y constante con los stakeholders y usuarios, cosa que la metodología tradicional no realizó. Por lo anterior, Agile UX recibió la calificación más alta.
- **Nivel de contacto con stakeholders:** gracias a la naturaleza iterativa y enfocada a la experiencia del usuario, el nivel de contacto con stakeholders de Agile UX fue bastante grande. Además, el agregado de realizarse el proceso dentro de un trabajo de grado, potenció ese nivel de contacto con stakeholders. Por otro lado, la metodología también contó con un buen nivel de contacto con stakeholders, pero se queda corto con respecto al proceso realizado mediante Agile UX. Por lo anterior, Agile UX recibió la nota más alta y la metodología tradicional, la calificación intermedia.

- **Nivel de contacto con usuarios:** sucedió lo mismo que el criterio anterior, además de que el mayor contacto que tuvieron con usuarios en el proceso de la metodología tradicional se limitó a un formulario enviado mediante correo electrónico y redes sociales, no hubo contacto humano directo, como en el caso de Agile UX, que por las circunstancias actuales del mundo, se tuvo que realizar mediante videoconferencias, pero igualmente fueron encuentros cara a cara digitales. Por las razones anteriores, Agile UX obtuvo la calificación más alta.
- **Cantidad de reuniones con stakeholders:** dada la naturaleza enfocada a la experiencia de usuario y el contexto de trabajo de grado en el cual se encontraba el proyecto, se realizaron muchas más reuniones con stakeholders en el proceso de Agile UX que en el realizado mediante metodología tradicional. Además estas reuniones fueron muy valiosas para la validación de requisitos. Por lo tanto, Agile UX obtuvo la calificación más alta.
- **Cantidad de reuniones con usuarios:** sucede lo mismo que en el criterio anterior, ya que debido a la naturaleza iterativa y centrada en la experiencia del usuario, Agile UX requirió de mucho contacto directo con los usuarios, mientras que el contacto del proceso de la metodología tradicional fue un poco más limitado, por lo tanto es más seguro decir que se obtuvo mayor retroalimentación y validación en el proceso de Agile UX. Por lo anterior, Agile UX obtuvo la calificación más alta.

6.4. Sesgos del análisis

Como se mencionó en secciones anteriores del documento, para los procesos de calificación y selección de metodologías se trabajó en conjunto con el director del trabajo, con el fin de validar las calificaciones y justificaciones dadas, esto teniendo en cuenta que la realización del proyecto fue de manera individual y no se cuentan con un equipo para realizar dichos procesos.

Teniendo en cuenta lo anterior, tampoco se buscó tener una opinión de expertos en las temáticas de metodologías de desarrollo o de los procesos de ingeniería de requisitos, lo cual hubiera podido generar un análisis más profundo gracias a su experiencia.

Adicionalmente, otro de los sesgos que se pudo generar, está relacionado con la población que fue entrevistada, ya que, como se explica en la sección de entrevistas, todos los sujetos entrevistados, eran personas pertenecientes de una forma u otra a la Pontificia Universidad Javeriana Cali, lo cual no permitió obtener una opinión externa, ya que la aplicación tratada durante el proceso, no solo va dirigida a personas de la institución, también para personas externas interesadas en conocer la información de los posgrados de la facultad de ingeniería y ciencias. Esto se ve como un problema debido a que las personas que fueron entrevistadas dieron sus puntos de vista, en muchos casos, relacionando los prototipos de la aplicación con las páginas institucionales de la universidad, lo cual pudo generar cierto nivel de preferencia y comodidad en algunos casos, situación que se podía mitigar si se entrevistaran a personas no relacionadas de ninguna forma a

la institución o sencillamente personas que nunca hubieran tenido contacto con las páginas web ni aplicaciones oficiales de la institución.

6.5. Conclusión del análisis comparativo

Después de realizar la justificación de cada criterio, y realizar la suma de los puntajes obtenidos en el cuadro 6.1 (Agile UX: 81 y metodología tradicional: 63), se pudo observar que la metodología mejor valorada y mejor respaldada es Agile UX, por lo tanto, la hace la metodología más adecuada para realizar el proceso de ingeniería de requisitos de aplicaciones web similares al caso de estudio tratado en este proyecto, lo cual tiene mucho sentido, ya que uno de los aspectos más importantes de las aplicaciones web, es la interacción [29] que mantienen los usuarios con las mismas, ya que esto puede generar un mayor nivel de uso por parte de ellos, que es lo que principalmente se busca al desarrollar y lanzar una aplicación nueva. En pocas palabras, la interacción del usuario con la aplicación se refiere a la experiencia del usuario al utilizarla, lo cual es el punto más importante manejado por la metodología Agile UX y la hace la más adecuada para este tipo de tareas.

Conclusiones

En conclusión, se puede decir que se realizó el proceso de análisis comparativo de manera exitosa, ya que se evaluaron aspectos importantes que deben tener en cuenta los procesos de ingeniería de requisitos. y también, se obtuvieron los resultados esperados, ya que la idea inicial era que alguno de los dos procesos de ingeniería de requisitos fuera seleccionado como el más adecuado para aplicaciones web similares a la tratada durante este proyecto, y gracias a las calificaciones de los criterios tenidos en cuenta para hacer el comparativo, la metodología seleccionada en las etapas tempranas del proyecto, Agile UX resultó ser la más adecuada.

Por otro lado, se pudo comprobar la que una metodología basada en la experiencia de usuario, en este caso Agile UX, genera un gran nivel de interacción entre el equipo encargado de realizar el proyecto y los stakeholders y usuarios, lo que garantizaba una comunicación más directa y completa, una inmersión por parte de los usuarios mucho más grande, lo cual permitió alcanzar un nivel más alto de cercanía y familiaridad en los prototipos que se desarrollaron.

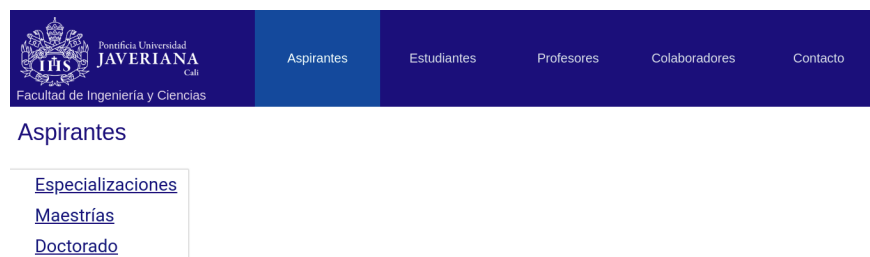
También se pudo evidenciar que la utilización de criterios bien definidos para realizar un proceso comparativo, permite llegar a resultados mucho más objetivos y robustos, lo que garantiza un alto índice de éxito al momento de tomar decisiones basadas en dichos resultados.

Además, se evidenció que ajustar el proceso propuesto, Agile UX, al proyecto, generó el resultado esperado, que era principalmente, demostrar que este era más adecuado que el proceso de la metodología tradicional.

Finalmente, se comprobó que entre más contacto se tiene con los usuarios, se obtuvo una mayor cercanía y familiaridad con los mismos, lo cual a su vez permitió generar una comunicación más directa y obtener requisitos muchos más aterrizados y útiles para efectos de la aplicación web.

Anexos prototipo 1

A.1. Pantalla de aspirantes



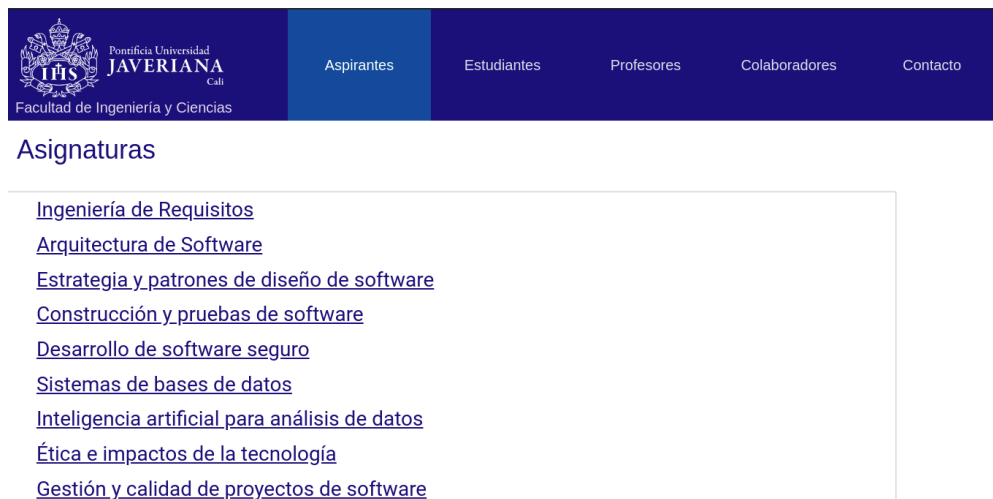
A.2. Pantalla de especializaciones para aspirantes



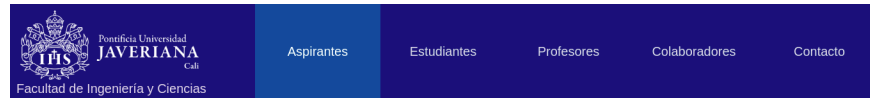
A.3. Pantalla de Especialización en Ingeniería de Software para aspirantes



A.4. Pantalla de asignaturas de la Especialización en Ingeniería de Software para aspirantes



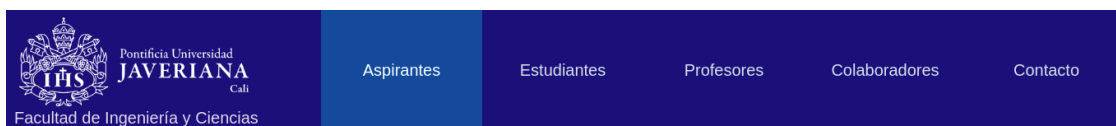
A.5. Pantalla de asignatura específica de la Especialización en Ingeniería de Software para aspirantes



Inteligencia artificial para el análisis de datos

Descripción	Contenidos
<p>El curso de Inteligencia Artificial para Análisis de Datos recopila métodos y técnicas relacionadas con el proceso de análisis de datos basado en el aprendizaje automático. Este proceso incluye el entendimiento y preparación de los datos, la aplicación de técnicas de inteligencia artificial para la construcción de modelos que capturan patrones recurrentes, y la evaluación de los modelos. En el curso se estudian los algoritmos de aprendizaje automático y los problemas más comunes que surgen al realizar una aplicación para una tarea específica.</p>	

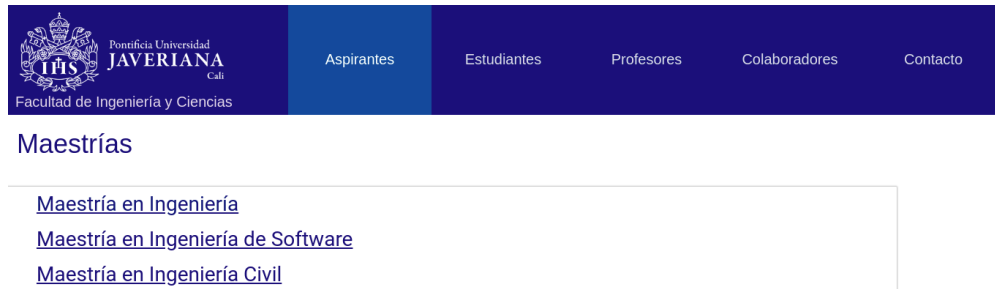
A.6. Pantalla de estudiantes



Estudiantes

- [Especializaciones](#)
- [Maestrías](#)
- [Doctorado](#)

A.7. Pantalla de maestrías para estudiantes



A.8. Pantalla de Maestría en Ingeniería de Software para estudiantes



A.9. Pantalla de asignaturas de la Maestría en Ingeniería de Software para estudiantes

The screenshot shows the top navigation bar of the website. On the left is the logo of Pontificia Universidad Javeriana, Cali, Facultad de Ingeniería y Ciencias. The navigation menu includes 'Aspirantes', 'Estudiantes', 'Profesores', 'Colaboradores', and 'Contacto'. Below the navigation bar, the page title is 'Asignaturas'. There are two columns of course links:

- [Ingeniería de Requisitos](#)
- [Arquitectura de Software](#)
- [Estrategia y patrones de diseño de software](#)
- [Construcción y pruebas de software](#)
- [Desarrollo de software seguro](#)
- [Sistemas de bases de datos](#)
- [Inteligencia artificial para análisis de datos](#)
- [Ética e impactos de la tecnología](#)
- [Gestión y calidad de proyectos de software](#)

- [Procesamiento distribuido de datos](#)
- [Arquitecturas de software en la nube](#)
- [El Negocio del Software](#)
- [Arquitecturas de integración](#)
- [Líneas de productos de software](#)

A.10. Pantalla de asignatura específica de la Maestría en Ingeniería de Software para estudiantes

The screenshot shows the top navigation bar with the same logo and menu as in A.9. Below the navigation bar, the page title is 'Líneas de productos de software'. There are two main content areas:

- Novedades**: A large light blue rectangular area intended for news or updates.
- Contenidos**: A smaller light blue rectangular area intended for course content.

A.11. Pantalla de login para profesores

Prototipo de la pantalla de login para profesores. El encabezado muestra el logo de la Pontificia Universidad JAVERIANA Cali y la Facultad de Ingeniería y Ciencias. El menú de navegación incluye: Aspirantes, Estudiantes, Profesores, Colaboradores y Contacto. El formulario de login contiene:

- Placeholder
- Placeholder
- Iniciar

A.12. Pantalla menú para profesores

Prototipo de la pantalla de menú para profesores. El encabezado muestra el logo de la Pontificia Universidad JAVERIANA Cali y la Facultad de Ingeniería y Ciencias. El menú de navegación incluye: Aspirantes, Estudiantes, Profesores, Colaboradores y Contacto. El contenido principal muestra:

BIENVENIDO(A) XXXXX XXXXXXXX

- Cursos
- Información personal
- Eventos

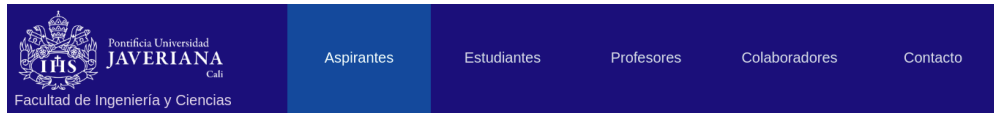
A.13. Pantalla de cursos para profesores

The screenshot shows a navigation bar at the top with the logo of Pontificia Universidad JAVERIANA Cali and the Facultad de Ingeniería y Ciencias. The navigation menu includes 'Aspirantes', 'Estudiantes', 'Profesores', 'Colaboradores', and 'Contacto'. Below the navigation bar, there is a section titled 'Cursos' containing three placeholder lines of text: 'XXXXXX XXXXXXXXXXXX', 'XXXXXXXXXXXXXXXXXXXXX', and 'XXXXXXXXXXXXXXXXXXXXX'.

A.14. Pantalla de curso específico para profesores

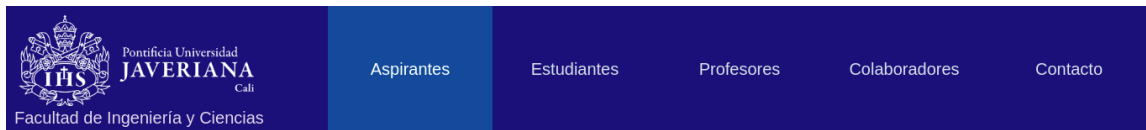
The screenshot shows a navigation bar similar to the previous one, but with 'Profesores' highlighted. Below the navigation bar, the text 'Curso: XXXXXX XXXXXXXXXXXX' is displayed. The main content area consists of three light blue rectangular buttons: 'Agregar entrada', 'Agregar trabajo destacado', and 'Horario'.

A.15. Pantalla de login para colaboradores



Colaboradores

A.16. Pantalla menú para colaboradores



BIENVENIDO(A) XXXXX XXXXXXX

Agregar post

Trabajos destacados

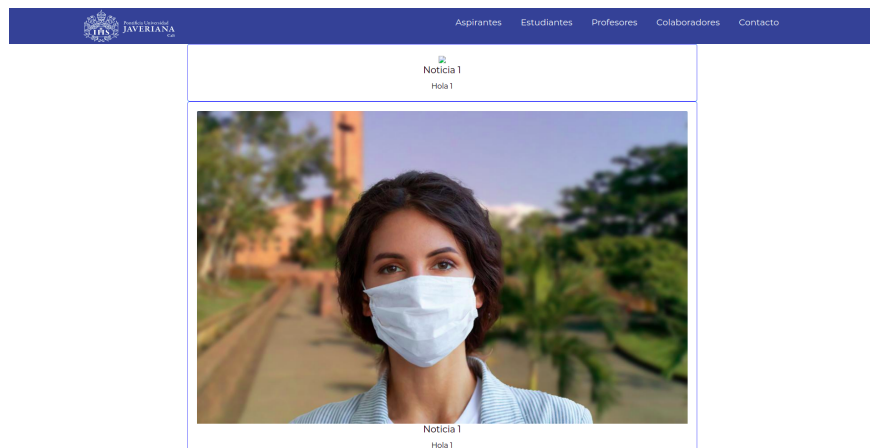
Formatos

A.17. Pantalla de crear noticia para colaboradores

The screenshot shows a web interface for creating a post. At the top, there is a dark blue navigation bar with the university logo and name on the left, and menu items 'Aspirantes', 'Estudiantes', 'Profesores', 'Colaboradores', and 'Contacto' on the right. The 'Colaboradores' item is highlighted. Below the navigation bar, the heading 'Crear post' is displayed. Underneath, there is a 'Título' label followed by a text input field. Below that is a 'Post' label followed by a larger text area for the main content. At the bottom center, there is a blue button labeled 'Publicar'.

Anexos prototipo 2

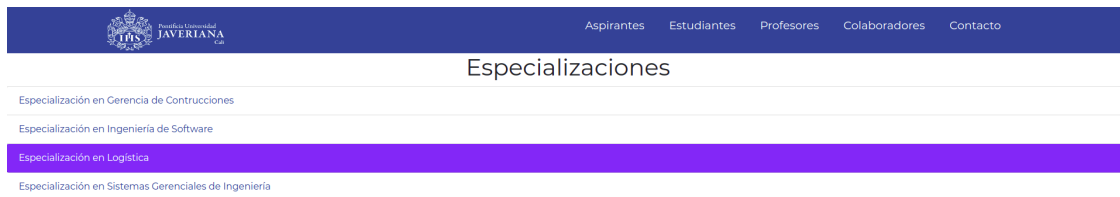
B.1. Pantalla de inicio



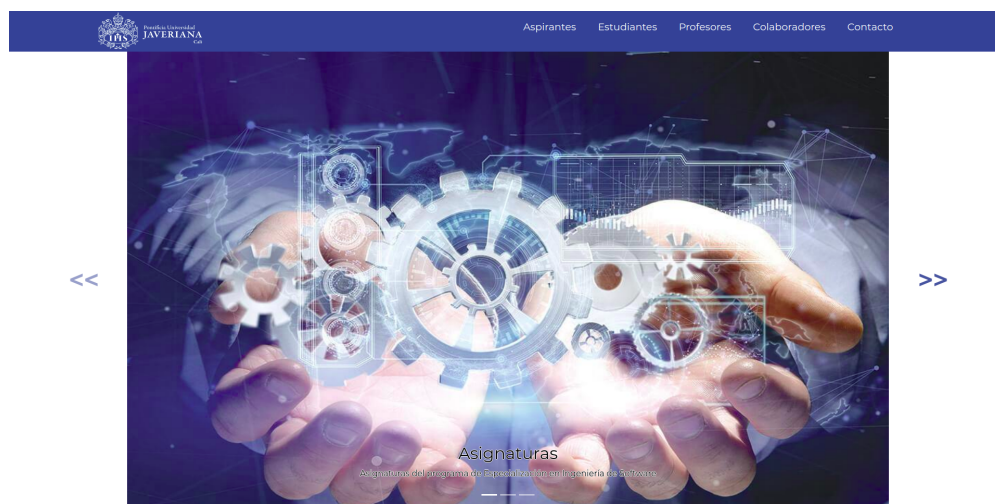
B.2. Pantalla de aspirantes



B.3. Pantalla de especializaciones para aspirantes

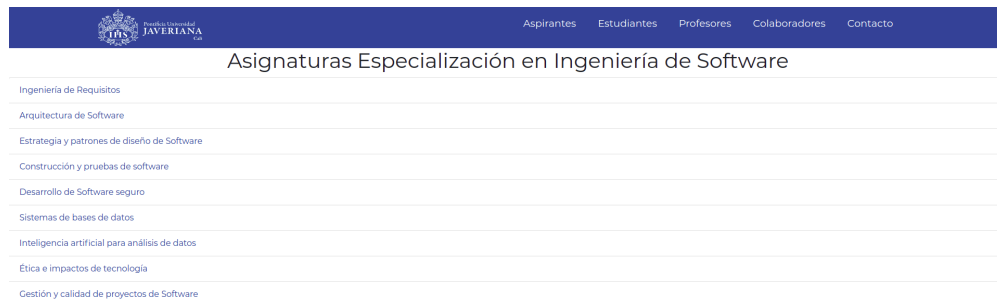


B.4. Pantalla de Especialización en Ingeniería de Software para aspirantes



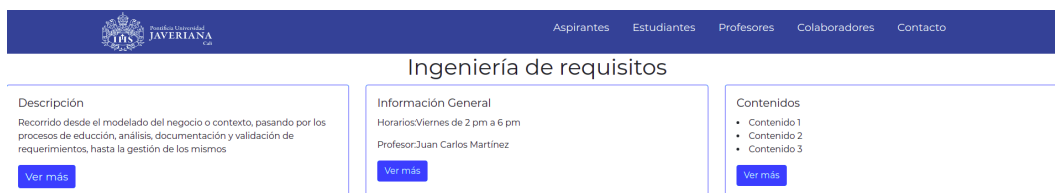
B.5. Pantalla de asignaturas de la Especialización en Ingeniería de Software para aspirantes\$7

B.5. Pantalla de asignaturas de la Especialización en Ingeniería de Software para aspirantes



Asignaturas Especialización en Ingeniería de Software	
Ingeniería de Requisitos	
Arquitectura de Software	
Estrategia y patrones de diseño de Software	
Construcción y pruebas de software	
Desarrollo de Software seguro	
Sistemas de bases de datos	
Inteligencia artificial para análisis de datos	
Ética e impactos de tecnología	
Gestión y calidad de proyectos de Software	

B.6. Pantalla de asignatura específica de la Especialización en Ingeniería de Software para aspirantes

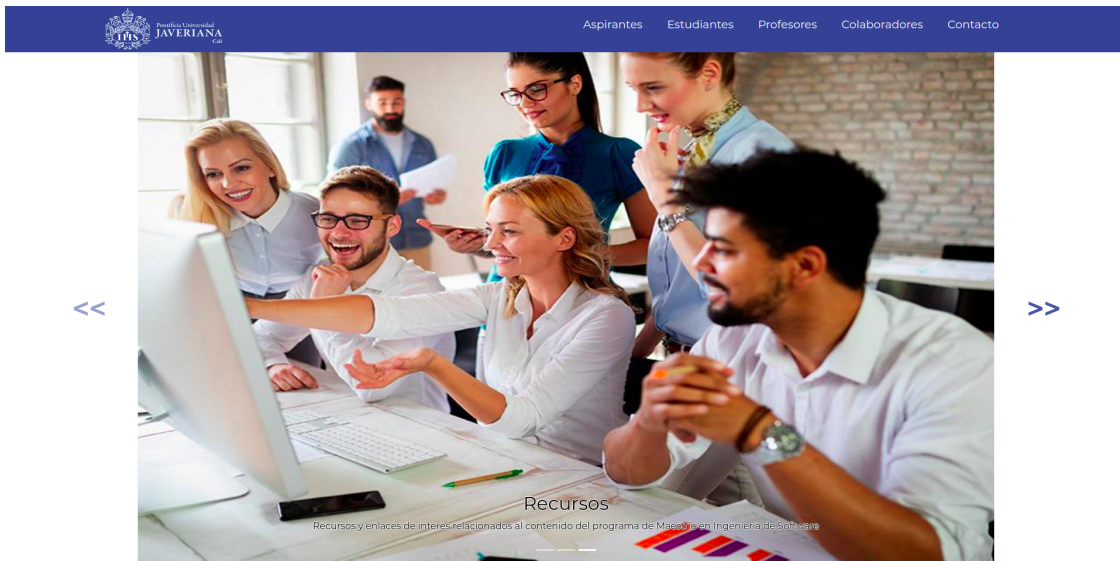


Ingeniería de requisitos		
Descripción Recorrido desde el modelado del negocio o contexto, pasando por los procesos de educación, análisis, documentación y validación de requerimientos, hasta la gestión de los mismos Ver más	Información General Horarios:Viernes de 2 pm a 6 pm Profesor:Juan Carlos Martínez Ver más	Contenidos <ul style="list-style-type: none">• Contenido 1• Contenido 2• Contenido 3 Ver más

B.7. Pantalla de maestrías para estudiantes



B.8. Pantalla de Maestría en Ingeniería de Software para estudiantes



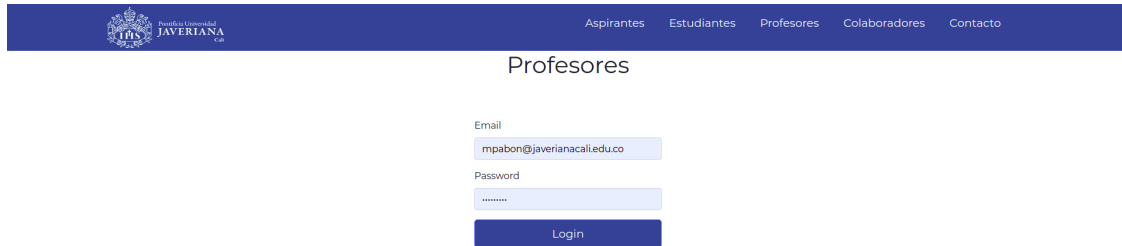
B.9. Pantalla de asignaturas Maestría en Ingeniería de Software para estudiantes



B.10. Pantalla de asignatura específica Maestría en Ingeniería de Software para estudiantes

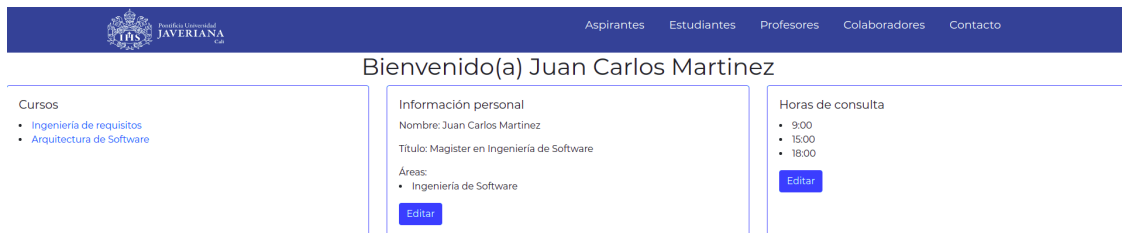


B.11. Pantalla de login para profesores



The screenshot shows the login page for teachers. At the top, there is a dark blue navigation bar with the logo of the Universidad Javeriana on the left and menu items: Aspirantes, Estudiantes, Profesores, Colaboradores, and Contacto. Below the navigation bar, the word "Profesores" is centered. The login form consists of three main elements: an "Email" field containing "mpabon@javerianacali.edu.co", a "Password" field with masked characters ".....", and a blue "Login" button.

B.12. Pantalla menú para profesores



The screenshot shows the menu page for teachers. At the top, there is a dark blue navigation bar with the logo of the Universidad Javeriana on the left and menu items: Aspirantes, Estudiantes, Profesores, Colaboradores, and Contacto. Below the navigation bar, the text "Bienvenido(a) Juan Carlos Martinez" is centered. The page is divided into three main sections:

- Cursos:** A list of courses including "Ingeniería de requisitos" and "Arquitectura de Software".
- Información personal:** A section containing the name "Nombre: Juan Carlos Martinez", the title "Titulo: Magister en Ingeniería de Software", and the area "Áreas: Ingeniería de Software". There is an "Editar" button below this section.
- Horas de consulta:** A section containing a list of consultation hours: "9:00", "15:00", and "18:00". There is an "Editar" button below this section.

B.13. Pantalla curso específico para profesores

Universidad JAVERIANA

Aspirantes Estudiantes Profesores Colaboradores Contacto

Curso: Ingeniería de requisitos

Agregar entrada
Agregar una noticia sobre la actualidad del curso
[Agregar](#)

Agregar trabajo destacado
Agregar un trabajo sobresaliente relacionado al curso para que los estudiantes puedan acceder al mismo.
[Agregar](#)

Horario
Viernes de 2 pm a 6 pm

B.14. Pantalla de agregar noticia para profesores

Universidad JAVERIANA

Aspirantes Estudiantes Profesores Colaboradores Contacto

Agregar noticia

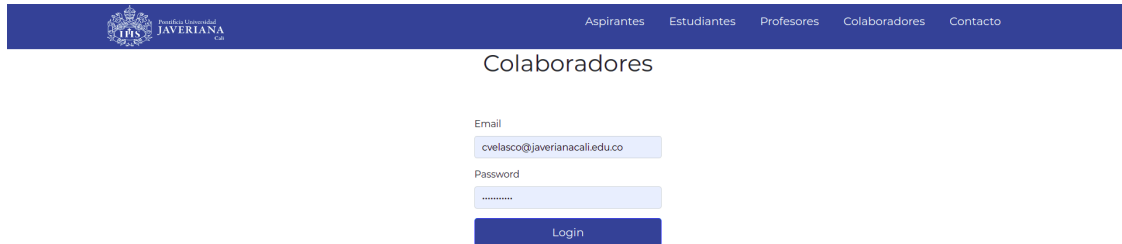
Titulo

Descripción

[Agregar](#)

[Atrás](#)

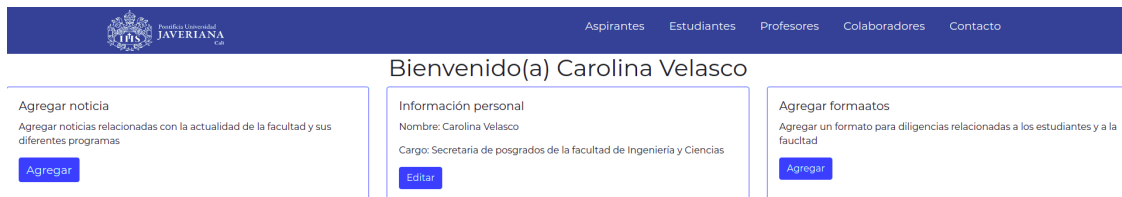
B.15. Pantalla de login para colaboradores



The screenshot shows a web interface for the login of collaborators. At the top, there is a dark blue navigation bar with the logo of the Pontificia Universidad JAVERIANA on the left and a menu with the items: Aspirantes, Estudiantes, Profesores, Colaboradores, and Contacto. Below the navigation bar, the page title is "Colaboradores". The main content area contains a login form with the following elements:

- An "Email" label above a text input field containing "cvelasco@javerianacali.edu.co".
- A "Password" label above a password input field with masked characters ".....".
- A dark blue "Login" button below the password field.

B.16. Pantalla menú para colaboradores



The screenshot shows a web interface for the menu of collaborators. At the top, there is a dark blue navigation bar with the logo of the Pontificia Universidad JAVERIANA on the left and a menu with the items: Aspirantes, Estudiantes, Profesores, Colaboradores, and Contacto. Below the navigation bar, the page title is "Bienvenido(a) Carolina Velasco". The main content area is divided into three columns, each with a title and a description, and a button:

- Column 1:** Title "Agregar noticia", description "Agregar noticias relacionadas con la actualidad de la facultad y sus diferentes programas", and a blue "Agregar" button.
- Column 2:** Title "Información personal", description "Nombre: Carolina Velasco" and "Cargo: Secretaria de posgrados de la facultad de Ingenieria y Ciencias", and a blue "Editar" button.
- Column 3:** Title "Agregar formatos", description "Agregar un formato para diligencias relacionadas a los estudiantes y a la facultad", and a blue "Agregar" button.

B.17. Pantalla de agregar noticia para colaboradores



The screenshot shows a web interface for adding a news item. At the top, there is a dark blue navigation bar with the Javeriana logo on the left and menu items: Aspirantes, Estudiantes, Profesores, Colaboradores, and Contacto. Below the navigation bar, the page title is 'Agregar noticia'. The form consists of three main sections: 1. 'Titulo' (Title) with a single-line text input field. 2. 'Descripción' (Description) with a larger multi-line text area. 3. 'Imagen' (Image) with a file selection button labeled 'Choose File' and the text 'No file chosen'. At the bottom of the form, there are two buttons: a blue 'Agregar' (Add) button and a dark blue 'Atrás' (Back) button.

Bibliografía

- [1] A. Virrueta Mendez, “Metodologías de desarrollo de software.” [Online] Available: https://www.ecured.cu/Metodologias_de_desarrollo_de_Software, 2010.
- [2] Northware, “Proceso de Diseño UI/UX en Desarrollo de Software y Apps - Northware.” [Online] Available: <https://www.northware.mx/2018/02/07/pocesode-diseno-ui-ux-en-desarrollo-de-software-y-apps/>, 2018.
- [3] OnDesarrollo, “¿Por qué es importante el Desarrollo Web? « OnDesarrollo.” [Online] Available: <https://ondesarrollo.com/por-que-es-importante-el-desarrollo-web/>, 2015.
- [4] K. Pohl and C. Rupp, *Requirements Engineering Fundamentals*. rockynook, 2015.
- [5] H. Tokkonen and P. Saariluoma, “How user experience is understood?,” *Proceedings of 2013 Science and Information Conference, SAI 2013*, pp. 791–795, 2013.
- [6] Platzi, “Diseño de Experiencia de Usuario (UX).” [Online] Available: <https://platzi.com/disenou-ux/>, 2018.
- [7] ISO25000, “Usabilidad.” [Online] Available: <https://iso25000.com/index.php/normas-iso-25000/iso-25010/23-usabilidad>, 2011.
- [8] L. R. Vijayarathy and C. W. Butler, “Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?,” *IEEE Software*, vol. 33, no. 5, pp. 86–94, 2016.
- [9] Wikipedia, “Desarrollo ágil de software - Wikipedia, la enciclopedia libre.” [Online] Available: https://es.wikipedia.org/wiki/Desarrollo_ágil_de_software, 2020.
- [10] D. Callele, E. Neufeld, and K. Schneider, “An introduction to experience requirements,” in *Proceedings of the 2010 18th IEEE International Requirements Engineering Conference, RE2010*, pp. 395–396, 2010.
- [11] P. C. Anitha and B. Prabhu, “Integrating requirements engineering and user experience design in Product life cycle Management,” in *2012 1st International Workshop on Usability and Accessibility Focused Requirements Engineering, UsARE 2012 - Proceedings*, pp. 12–17, 2012.
- [12] P. J. Denning, “Design thinking,” in *Communications of the ACM*, vol. 56, pp. 29–31, dec 2013.
- [13] F. Dobrigkeit and D. De Paula, “Design thinking in practice: Understanding manifestations of design thinking in software engineering,” in *ESEC/FSE 2019 - Proceedings of the 2019 27th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 1059–1069, Association for Computing Machinery, Inc, aug 2019.

- [14] G. Jurca, T. D. Hellmann, and F. Maurer, "Integrating agile and user-centered design: A systematic mapping and review of evaluation and validation studies of agile-UX," in *Proceedings - 2014 Agile Conference, AGILE 2014*, pp. 24–32, Institute of Electrical and Electronics Engineers Inc., sep 2014.
- [15] A. Srivastava, S. Bhardwaj, and S. Saraswat, "SCRUM model for agile methodology," in *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017*, vol. 2017-January, pp. 864–869, Institute of Electrical and Electronics Engineers Inc., dec 2017.
- [16] N. Pillay and J. Wing, "Agile UX: Integrating good UX development practices in Agile," in *2019 Conference on Information Communications Technology and Society, ICTAS 2019*, Institute of Electrical and Electronics Engineers Inc., apr 2019.
- [17] T. Batova, "Extended abstract: Lean UX and innovation in teaching," in *IEEE International Professional Communication Conference*, vol. 2016-November, Institute of Electrical and Electronics Engineers Inc., nov 2016.
- [18] L. A. Liikkanen, H. Kilpiö, L. Svan, and M. Hiltunen, "Lean UX - The next generation of user-centered Agile development?," in *Proceedings of the NordiCHI 2014: The 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, pp. 1095–1100, Association for Computing Machinery, Inc, oct 2014.
- [19] A. Pérez, "Stakeholders: ejemplos para entender el concepto." [Online] Available: <https://obsbusiness.school/es/blog-investigacion/marketing-y-comunicacion/stakeholders-ejemplos-para-entender-el-concepto>, 2017.
- [20] A. Zeaaraoui, Z. Bougroun, M. G. Belkasmi, and T. Bouchentouf, "User stories template for object-oriented applications," in *2013 3rd International Conference on Innovative Computing Technology, INTECH 2013*, pp. 407–410, 2013.
- [21] R. G. Tamarit, "¿Qué es el Backlog? - Muy Agile." [Online] Available: <https://muyagile.com/que-es-el-backlog/>, 2019.
- [22] Proto.io, "Features - Proto.io." [Online] Available: <https://proto.io/en/features/>, 2020.
- [23] React.js, "React – Una biblioteca de JavaScript para construir interfaces de usuario." [Online] Available: <https://es.reactjs.org/>.
- [24] Node.js, "Acerca — Node.js." [Online] Available: <https://nodejs.org/es/about/>.
- [25] MongoDB, "La base de datos líder del mercado para aplicaciones modernas — MongoDB." [Online] Available: <https://www.mongodb.com/es>.
- [26] L. S. Osorio Ortiz, H. W. Victoria Sandoval, and C. A. Neira Guzmán, "Documento de especificación de requisitos de software: Sistema de divulgación académica de los posgrados de la facultad de ingeniería," tech. rep., Pontificia Universidad Javeriana Cali, Sept. 2019.

-
- [27] Universidad Icesi, “SRS / ERS Especificación de requerimientos de software [Wiki LISA].” [Online] Available: https://www.icesi.edu.co/departamentos/tecnologias_informacion_comunicaciones/proyectos/lisa/home/analisis/srs/srs, 2010.
- [28] Z. Zhang, “Effective Requirements Development-A Comparison of Requirements Elicitation Techniques,” tech. rep., Tampere University, 2007.
- [29] J. Stephan, “Importance of UI UX Design in Building Web and Mobile Applications — by James Stephan — Medium.” [Online] Available: <https://medium.com/@iamjamesstephan/importance-of-ui-ux-design-in-building-web-and-mobile-applications-98099966e9ea>, 2019.