

**Departamento de Ingeniería de Sistemas y Computación**  
**Semestre: 2020-2**

**Proyecto de Grado:**  
**Herramientas Para Apoyar Arquitectos de Negocio**

**Presentado por:**  
**Daniel Felipe Serrano Mora**

**Presentado a:**  
**Jorge Alberto Villalobos Salcedo**

**Bogotá, 14 de diciembre de 2020**

## Contenido

0 Resumen.....	3
1 Introducción .....	3
2 Descripción general.....	4
2.1 Objetivos .....	4
2.2 Antecedentes.....	4
2.3 Identificación del Problema y su Importancia.....	5
3 Diseño y Especificaciones.....	6
3.1 Definición del Problema.....	6
3.2 Especificaciones .....	6
3.3 Restricciones .....	9
4 Desarrollo del diseño.....	9
4.1 Recopilación de la Información.....	9
4.2 Prototipado.....	12
4.3 Detalles de diseño finales.....	13
5 Implementación .....	14
5.1 Descripción de la Implementación .....	18
5.2 Resultados esperados .....	21
6 Validación .....	21
6.1 Métodos.....	21
6.2 Validación de Resultados .....	21
7 Conclusiones.....	21
7.1 Discusión.....	21
7.2 Trabajo a Futuro.....	22
8 Referencias.....	23
Bibliografía .....	23
Anexos.....	24

## **0 RESUMEN**

El proyecto consiste en la realización de una aplicación web que permita a los arquitectos de negocio realizar diversos tipos de diagramas relevantes en su proceso de consultoría. Durante el proyecto, se utilizaron metodologías ágiles para conducir a un equipo de desarrollo y realizar la planeación y el seguimiento del proceso. La motivación principal para la realización de este trabajo es la falta de una herramienta especializada en el desarrollo de los diagramas de la arquitectura de negocio. Con este proyecto se propone que el usuario pueda tener todos los diagramas de un proyecto en la misma interfaz con el fin de poder analizar el negocio de una manera más rápida y fácil. Finalmente, se obtuvo como resultado una aplicación web que cumple con las funcionalidades y requerimientos establecidos en las etapas de diseño en el tiempo esperado.

## **1 INTRODUCCIÓN**

En el presente texto, se busca documentar el proceso de desarrollo del proyecto de grado, desde su etapa de planeación hasta la etapa final de despliegue. Se espera que, al finalizar el documento, el lector comprenda lo realizado y se familiarice con el proceso de desarrollo de un proyecto de software.

Con el presente proyecto de grado, se busca realizar una herramienta web para ayudar en la creación de los diversos modelos de la arquitectura de negocio. Actualmente, los Arquitectos de negocio no tienen una herramienta específica para visualizar sus diagramas y tablas. Esto ha llevado a que el proceso de definir las principales vistas de un negocio sea tedioso y se demore más tiempo del deseado. Por lo tanto, se espera que, con la nueva herramienta, entender el negocio y sus componentes se pueda realizar de una manera más efectiva, tanto en un entorno profesional como en uno académico.

Para el diseño de la solución en primer lugar se definieron las funcionalidades principales que debería tener el proyecto, así como su cronograma de implementación. Además, se realizó un prototipo de baja fidelidad con el fin de obtener retroalimentación y realizar ajustes rápidamente. Adicionalmente, se contó con un equipo de diseño con el que se definieron diversos elementos estéticos tales como la paleta de colores, la tipografía y detalles de la interfaz de usuario. En cuanto a la implementación, se contó con un equipo de desarrollo para que se realizara el proyecto. La herramienta seleccionada para realizarlo fue una página web en React

en la cual progresivamente se fueron añadiendo los diversos modelos por medio de Sprints.

Como resultados obtenidos se consiguió que la página desarrollada se desplegara en un servidor y pudiera ser accedida por dirección web. Además, se entregó un repositorio debidamente documentado en el cual se establecen los pasos a seguir para posiblemente extender el proyecto y se expone el código fuente.

El presente documento está organizado de la siguiente manera: En primer lugar, se explicará a mayor profundidad la problemática y se expondrán las necesidades de diseño iniciales. En segundo lugar, se explicará como fue el proceso de desarrollo e implementación del proyecto. Finalmente, se presentarán los resultados obtenidos y se concluirá.

Antes de continuar, es necesario agradecer al equipo de desarrollo: Allan Corinaldi, Daniella Arteaga y Juan Pablo Correa. Además, al equipo de diseño: María Camila Romero y Luis Gómez. Finalmente, gracias a Jorge Villalobos, el cual fue un excelente asesor de tesis.

## **2 DESCRIPCIÓN GENERAL**

### 2.1 Objetivos

El principal objetivo y entregable de este proyecto de grado consiste en una aplicación web, la cual debe estar debidamente implementada y documentada, siguiendo con los patrones de arquitectura recomendados. Además, la experiencia de usuario dentro de la aplicación deberá de ser favorable y la solución deberá tener una identidad consolidada. Dentro de la aplicación el usuario deberá ser capaz de realizar los siguientes modelos y sus respectivos cruces:

- Modelo de Actores
- Portafolio de Servicios
- Estructura del modelo de negocio
- Modelo Ontológico
- Mapa de capacidades

### 2.2 Antecedentes

Actualmente existen en el mercado diversas herramientas diseñadas para la realización de diagramas. Además, existen también recursos enfocados en realizar arquitecturas de mayor nivel profesional. A continuación, se evaluarán estas herramientas para revisar su importancia y aporte a las necesidades de estudiantes y consultores:

- Archi (<https://www.archimatetool.com/>): Esta herramienta está especializada en realizar diagramas de Arquitectura Empresarial. Actualmente, esta herramienta permite realizar diagramas de alto nivel y con un gran componente de detalle. Si bien esta herramienta puede ser ideal para los arquitectos empresariales, resulta difícil para personas con conocimiento de negocio realizar este tipo de diagramas debido a que Archi presenta un lenguaje extenso (Archimate 3.0).
- Bizagi (<https://www.bizagi.com/>): Esta herramienta está enfocada en la realización de modelos de procesos utilizando el lenguaje BPMN. Si bien es cierto que la herramienta es excelente para realizar este tipo de modelos, no está diseñada para el desarrollo de otro tipo de modelos, incluyendo los modelos de negocio.
- LucidChart (<https://www.lucidchart.com/pages/>): Esta herramienta web se enfoca en la realización de diagramas de cualquier tipo e incluye una funcionalidad de colaboración con otros usuarios. Las opciones de elementos que tiene el usuario para realizar diagramas son extensas, se pueden hacer desde organigramas hasta diagramas UML. Sin embargo, debido a su extensión, es posible mezclar elementos de diversos lenguajes y el diseño de los diagramas va primero antes del contenido del negocio.
- Otras Herramientas (Word, Excel y Powerpoint): A pesar de que estas herramientas no tienen como funcionalidad principal el desarrollo de modelos y diagramas, muchos estudiantes y consultores las utilizan para realizar sus propias tablas y diagramas libremente. Sin embargo, debido a que estas herramientas fueron diseñadas inicialmente para otros propósitos, no cumplen con un lenguaje ni conceptos definidos y es responsabilidad del usuario revisar la calidad del modelo.

### 2.3 Identificación del Problema y su Importancia

El principal problema consiste en la falta de una herramienta que apoye tanto en escenarios académicos como en ambientes profesionales de consultoría que se centre en las necesidades de los usuarios y que los ayude en la diagramación para que el usuario solamente tenga que aportar la información clave del negocio y no tener tanta preocupación por el diagrama.

Este problema es importante debido a que, en un ambiente académico, los diagramas del modelo de negocio son fundamentales para revisar si el estudiante puede aplicar los conceptos vistos a un caso de estudio y realizar los diagramas a partir del enunciado. En el caso de un ambiente de consultoría el modelo de negocio es el primer paso en el establecimiento de la arquitectura de negocio de una organización y ayuda a comprender qué es el negocio.

### 3 DISEÑO Y ESPECIFICACIONES

#### 3.1 Definición del Problema

El problema, concretamente consiste en la falta de una herramienta enfocada en el desarrollo de los diversos elementos del modelo de negocio.

#### 3.2 Especificaciones

A continuación, se presentan los requerimientos funcionales y no funcionales esenciales para el cumplimiento de los objetivos del proyecto:

##### Requerimientos Funcionales

Id	Nombre	Descripción	Entrada y parámetros	Resultado esperado
RF1	Registrar un usuario	Permitir que un usuario se registre en la aplicación	Nombre de usuario, correo y contraseña del usuario.	El usuario debe quedar registrado en la base de datos de la aplicación.
RF2	Login	Permitir que un usuario ya registrado ingrese a la aplicación.	Correo y contraseña del usuario.	Si las credenciales son válidas, el usuario entra en la aplicación, de lo contrario, se le notifica.
RF3	Crear Proyecto	Permitir que un usuario registrado pueda crear proyectos.	Hacer click en el botón de crear proyecto.	El proyecto ha sido creado y todos los modelos se encuentran en blanco y su

				estado es sin editar.
RF4	Editar Proyecto	Permitir cambiar el nombre de un proyecto.	Nuevo nombre.	El nombre del proyecto se cambia en la base de datos y se refleja en la interfaz.
RF5	Eliminar Proyecto	Permite eliminar un proyecto.	El usuario hace click en el botón de eliminar.	El proyecto y sus modelos se eliminan.
RF6	Editar Catálogo de actores	Permite la creación y edición del catálogo de actores. Permite agregar nuevos actores.	El usuario agrega o edita un actor.	Los cambios del modelo se reflejan en la interfaz y en la base de datos.
RF7	Editar Portafolio de Servicios	Permite la creación y edición del portafolio de servicios. Permite agregar nuevos servicios. Permite agregar nuevas operaciones.	El usuario edita el portafolio de servicios.	Los cambios del modelo se reflejan en la interfaz y en la base de datos.
RF8	Editar Modelo Ontológico	Permite la creación y edición del modelo ontológico. Permite agregar clases y sus atributos. Permite agregar relaciones. Permite agregar subclases.	El usuario edita el modelo ontológico.	Los cambios del modelo se reflejan en la interfaz y en la base de datos.
RF9	Editar Estructura del modelo de negocio	Permite la creación y edición de la estructura del modelo de negocio. Permite agregar canales. Permite agregar componentes.	El usuario edita la estructura del modelo de negocio.	Los cambios del modelo se reflejan en la interfaz y en la base de datos.

		Permite agregar un pool interno.		
RF10	Editar Mapa de Capacidades	Permite la creación y edición del mapa de capacidades. Permite agregar capacidades en cada uno de los 3 niveles.	El usuario edita el mapa de capacidades.	Los cambios del modelo se reflejan en la interfaz y en la base de datos.
RF11	Exportar Diagrama	Permite exportar lo creado en los diagramas de la aplicación web localmente al dispositivo del usuario.	Dentro de la vista de algún diagrama, el usuario hace click en exportar.	El diagrama es exportado en formato png.
RF12	Exportar Tabla	Permite exportar lo creado en las tablas de la aplicación web localmente al dispositivo del usuario.	Dentro de la vista de alguna tabla, el usuario hace click en exportar.	La tabla es exportada en formato pdf.

#### Requerimientos no funcionales

ID	Nombre	Descripción	Resultado esperado
RNF1	Autenticación	El usuario solamente debe ver sus proyectos y debe estar autenticado para realizarlo.	No se debe permitir el acceso a proyectos ni a modelos que no sean propiedad del usuario. Si el usuario cierra sesión, se deben dejar de mostrar los proyectos hasta que vuelva a iniciar sesión.
RNF2	Persistencia	Se deben guardar los proyectos y modelos a medida que el usuario los edite.	Si el usuario presiona guardar, la base de datos con los modelos se debe actualizar. Además, periódicamente se debe generar un guardado automático. Siempre que se guarde se debe notificar al usuario.



RNF 3	Manejo de Errores	Si el usuario trata de acceder a una url a la que no tenga acceso o que esté errónea, el sistema debe notificar.	El sistema muestra un mensaje de error que dice que el sitio al que está tratando de acceder no existe.
-------	-------------------	--	---

### 3.3 Restricciones

La principal restricción que se tiene es la restricción temporal debido a que el tiempo del proyecto total debe ser de un semestre. Además, hay otras restricciones técnicas como la cantidad de personas en el proyecto y las limitaciones del lenguaje de programación. Se decidió utilizar REACT para el desarrollo del proyecto debido a que este framework es el más conocido por el equipo de desarrollo.

## 4 DESARROLLO DEL DISEÑO

El proceso de diseño consistió en 3 etapas: Levantamiento de información, Prototipado y definición final de Diseño. En la etapa de levantamiento de información, se realizaron encuestas y entrevistas a estudiantes que vieron el curso de arquitectura empresarial con el fin de conocer la manera en la que actualmente realizaban los diagramas y tablas del modelo de negocio y, así, identificar sus principales puntos de dolor. Para la parte de prototipado, se partió de las funcionalidades esperadas de la aplicación para desarrollar un prototipo de baja fidelidad en el que se viera a grandes rasgos la navegación de la aplicación y su estructura. Finalmente, en la definición del diseño, se estableció, de la mano del equipo de diseño, la paleta de colores, la tipografía y la metáfora de diseño que se iba a utilizar en el proyecto final.

### 4.1 Recopilación de la Información

Para comenzar a plantear el diseño se realizó una encuesta a 33 estudiantes que vieron el curso de Arquitectura Empresarial en 2019 y en 2020. Las preguntas de la encuesta están orientadas a identificar la forma actual de diseño de diagramas y modelos y sus principales puntos de dolor.

Actualmente se evidencia que la mayoría de los estudiantes se siente conformes con la realización de modelos de la arquitectura de negocio, tal y como se evidencia en la Figura 1:

¿Qué tan fácil le es para usted realizar modelos de negocio? ((1) el portafolio de servicios, (2) la estructura del negocio y (3) el modelo ontológico)

33 respuestas

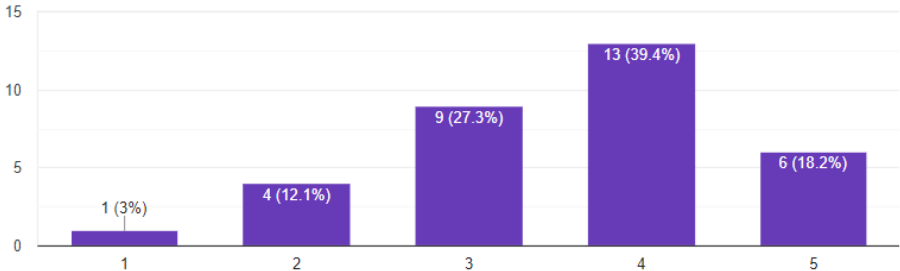


Figura 1. Estado actual de los estudiantes en la realización de modelos de negocio.

Entre las herramientas, Lucidchart es la más popular a la hora de realizar estos diagramas, principalmente por su componente colaborativo.

Otro dato interesante de la encuesta es el tiempo que los usuarios se demoran realizando estos diagramas, para la mayoría de los encuestados el tiempo es mayor a 2 horas:

¿Cuánto tiempo (aproximadamente) se tarda en realizar un modelo de negocio correctamente? ((1) el portafolio de servicios, (2) la estructura del negocio y (3) el modelo ontológico)

31 respuestas

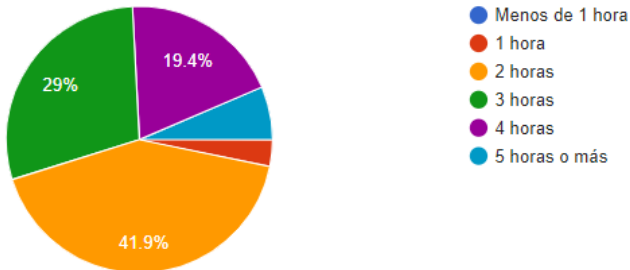


Figura 2. Tiempo requerido para realizar el modelo de negocio.

Además, otro modelo en el que los estudiantes presentaban más dificultades a la hora de desarrollar es el mapa de capacidades de un negocio. Si bien este modelo

no hace parte del modelo de negocio, su información puede ayudar a detallar las acciones que realiza la compañía. Esta información se evidencia en la Figura 3.

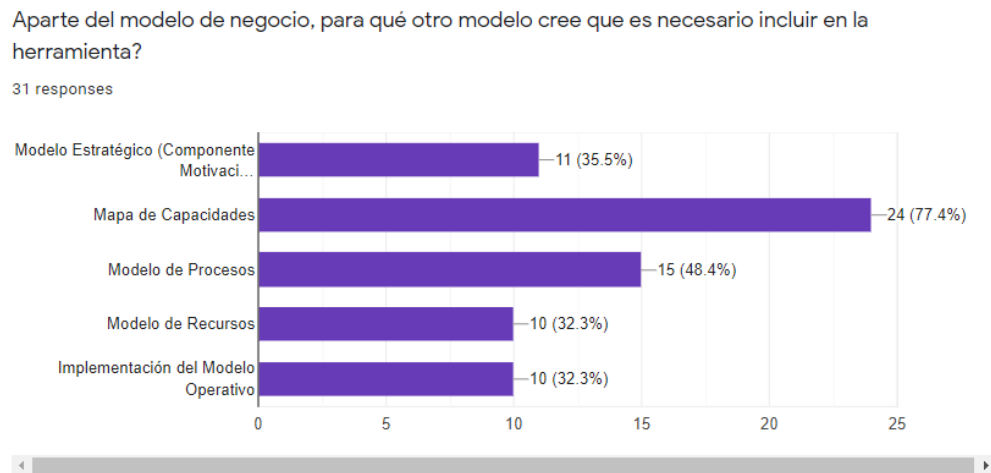


Figura 3. Descripción de otras herramientas adicionales.

Teniendo en cuenta los resultados anteriores, se realizaron entrevistas a estudiantes para profundizar en los problemas que presentan actualmente y revisar las posibles funcionalidades de la nueva herramienta. De acuerdo con esas entrevistas, se identificaron los siguientes aspectos:

- El lenguaje que se debe manejar debe ser simple, sencillo e intuitivo. Los estudiantes y consultores no deben tener que aprender lenguajes adicionales a los que manejan. Por esta razón, se decidió utilizar un lenguaje que ya estuviera establecido para cada modelo que fuera lo suficientemente simple para entender y lo suficientemente complejo para realizar diagramas avanzados.
- Dentro de la herramienta es necesario que se puedan realizar algunos elementos del modelo de negocio utilizando tablas y otros elementos del modelo de negocio utilizando diagramas. Para efectos de esta solución, se decidió utilizar tablas para el catálogo de actores, el portafolio de servicios y el mapa de capacidades. Para el modelo ontológico y la estructura del modelo de negocio se decidió utilizar un lenguaje gráfico.
- La organización de los modelos dentro de la aplicación debería estar organizada por proyectos. Cada proyecto debe tener los diagramas y tablas necesarios para plantear una arquitectura de negocio.
- Como funcionalidad de valor adicional, es fundamental que los componentes usados en un modelo del proyecto puedan ser utilizados para los demás modelos. Esto es necesario para que se mantenga una coherencia sintáctica

de los componentes y para que el usuario no tenga que ingresar información repetidamente y que sea redundante.

## 4.2 Prototipado

En cuanto a la parte de prototipado, primero se definió un prototipo de baja fidelidad que mostraba las principales funcionalidades que tendría la aplicación. Después de realizar ese prototipo, se definió un mockup con una interfaz más similar a la interfaz final de la aplicación. La principal función de estos prototipos es mostrar cómo sería la aplicación tanto al equipo de desarrollo como a los usuarios finales para tener retroalimentación en cuanto a su diseño.

En la siguiente figura se presenta una de las imágenes del prototipo inicial realizado. Este prototipo inicial se enfocaba en establecer la estructura de la herramienta y mostrar cómo sería el resultado esperado de las funcionalidades. Por esta razón, el prototipo no posee componentes estéticos de la interfaz.

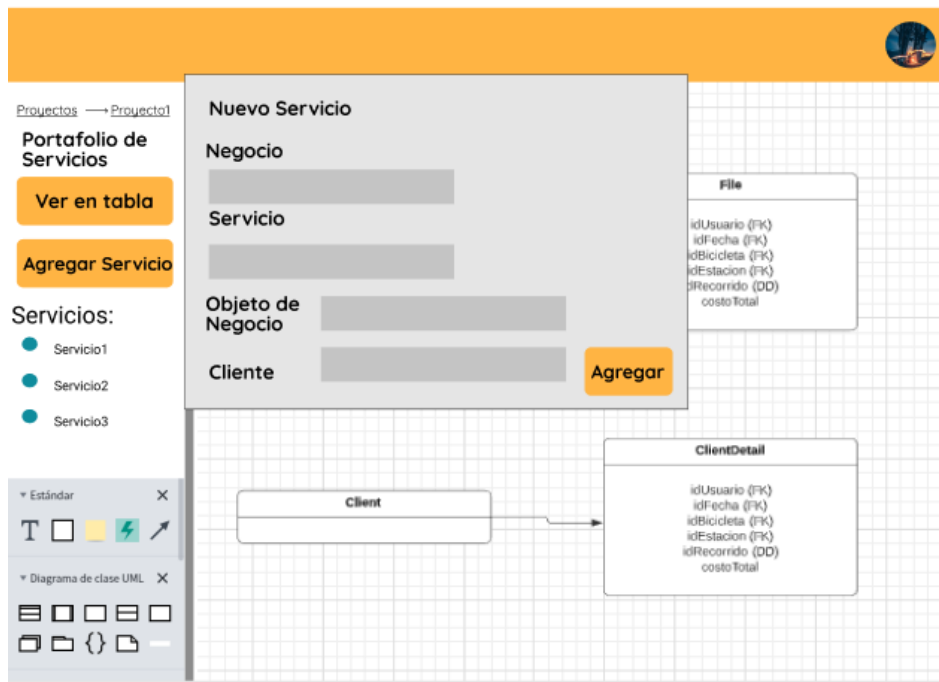


Figura 4. Vista del portafolio de servicios del prototipo inicial.

En la siguiente figura se puede evidenciar el mockup realizado. Este mockup se asemeja al resultado final en gran parte. Sin embargo, se decidió simplificar la

interfaz de edición para que solamente tuviera los editables mínimos necesarios para que los diagramas fueran funcionales.

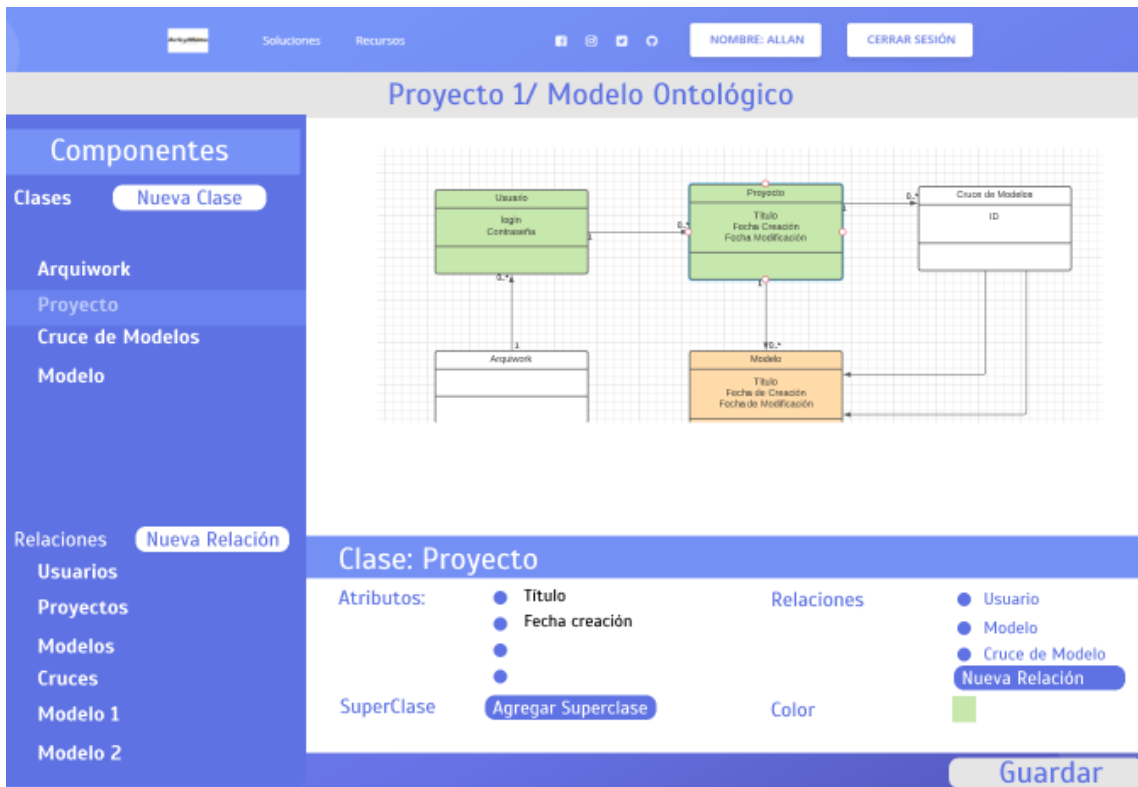


Figura 5. Vista del modelo ontológico en el mockup de la aplicación.

El resto de las vistas de los prototipos y el mockup se encuentran en los documentos adjuntos.

#### 4.3 Detalles de diseño finales

Finalmente, se realizó un diseño final en compañía del equipo de diseño de interfaz. Para realizar el análisis y la definición del estilo de la aplicación se tuvo en cuenta la intención de la aplicación, el usuario objetivo y el estilo de diseño de las aplicaciones seleccionadas. Como heurística de diseño web, se decidió utilizar Material UI para los componentes. Los principales resultados del análisis se encuentran en las siguientes figuras y el documento completo se encuentra en el anexo.

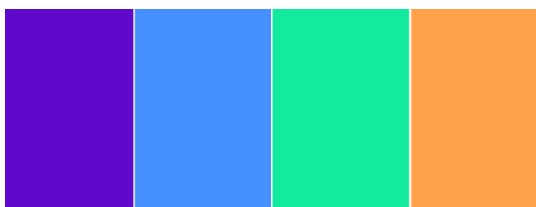


Figura 6. Definición de la paleta de colores



Figura 7. Tipografía definida para la aplicación.

Estos descubrimientos del estudio de diseño ayudaron a formar un 'Look & feel' en la aplicación web y a mejorar la experiencia del usuario en general.

## 5 IMPLEMENTACIÓN

Como se ha mencionado anteriormente, para la implementación se contó con un equipo de desarrollo el cual ayudó a construir la aplicación final. La solución final fue una aplicación web para la cual se utilizó el Framework web de React. Además, el lenguaje de programación utilizado fue JavaScript y se utilizó css para los estilos. Para la parte de los datos se definió la utilización de una base de datos documental y la base de datos seleccionada fue MongoDB. Debido a que era necesario permitir la creación y edición de diagramas, se incluyó una librería para realizar gráficos llamada mxGraph. Esta librería es de uso abierto y es utilizada por otras herramientas de diagramación tales como draw.io. Todo el código realizado fue subido constantemente a un repositorio en Github, en el que todo está debidamente documentado. Se le atribuyó al repositorio una licencia MIT para que sea libre de uso al público. Finalmente, para el despliegue de la aplicación se utilizó Heroku para montar un servidor con la aplicación y que ésta pueda ser accedida por cualquier usuario.

Con respecto a la arquitectura, se decidió optar por una arquitectura base cliente servidor en la cual los editables que realizara el usuario se fueran guardando periódicamente en la base de datos. Obtener diagramas y tablas ya creados se realiza bajo petición al servidor y requiere que el usuario haya iniciado sesión para garantizar la seguridad de sus proyectos.

En cuanto a otras herramientas utilizadas, cabe destacar que se utilizó el stack MERN, el cual es muy popular a la hora de construir aplicaciones web. Este conjunto de herramientas está compuesto de MongoDB para la persistencia, Express para la autenticación, React como framework principal y Node.js para la lógica del back-end.

El modelo de datos utilizado para manejar el proyecto y con el cual se realizó la persistencia en MongoDB fue el siguiente:

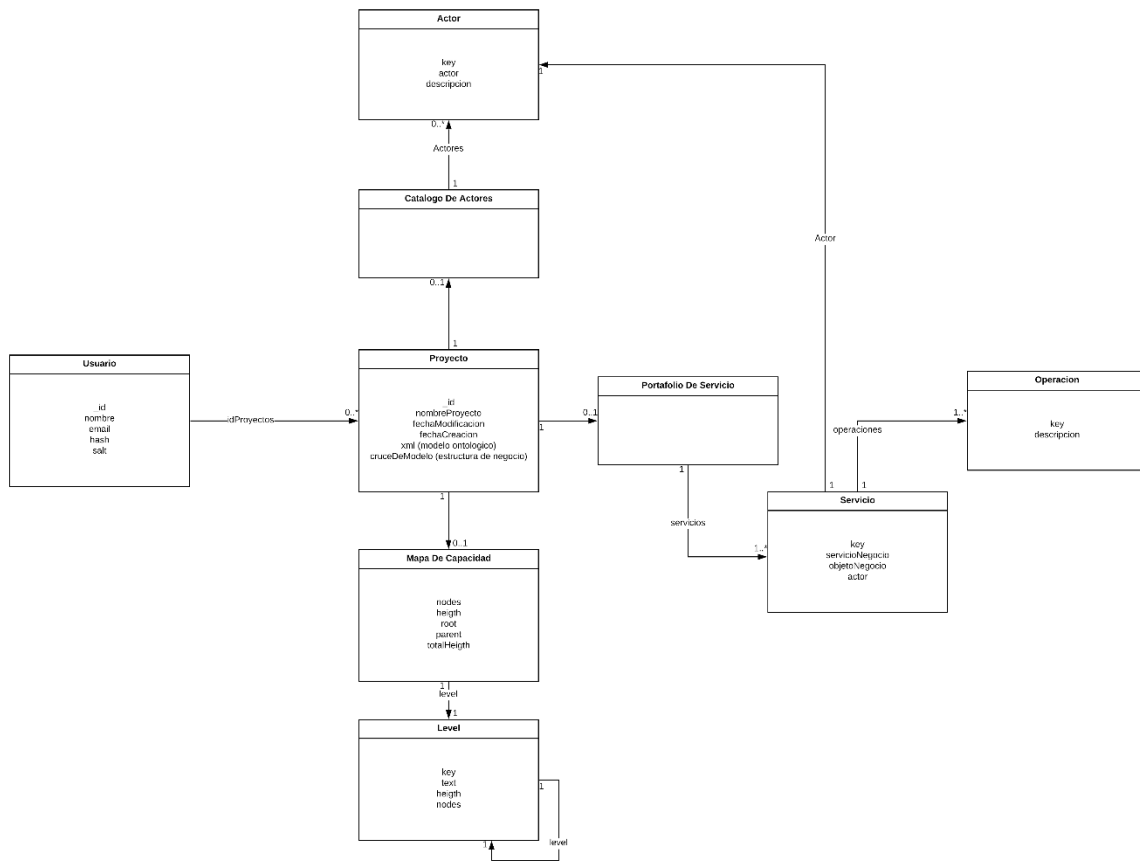


Figura 8. Modelo de datos

El resultado final de la aplicación se encuentra anexo. Sin embargo, en las siguientes figuras se presentan las funcionalidades principales:



Figura 9. Página principal con los proyectos del usuario.



Figura 10. Página con el catálogo de actores.





Figura 11. Página con el portafolio de servicios. Solamente es posible seleccionar a los actores creados en el catálogo de actores.

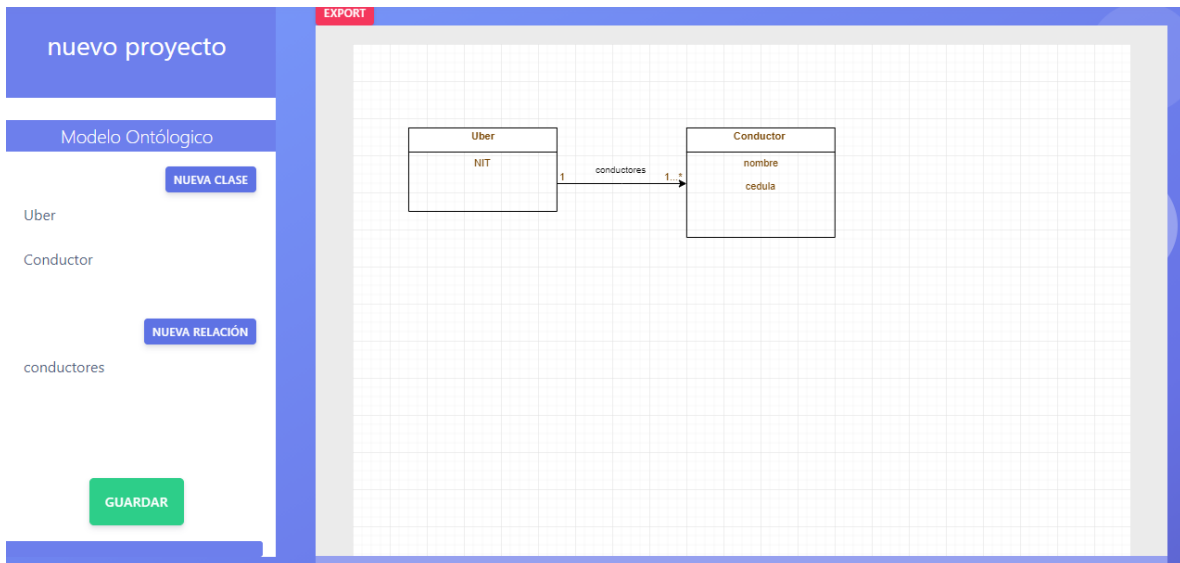


Figura 12. Página con el editor del modelo ontológico.

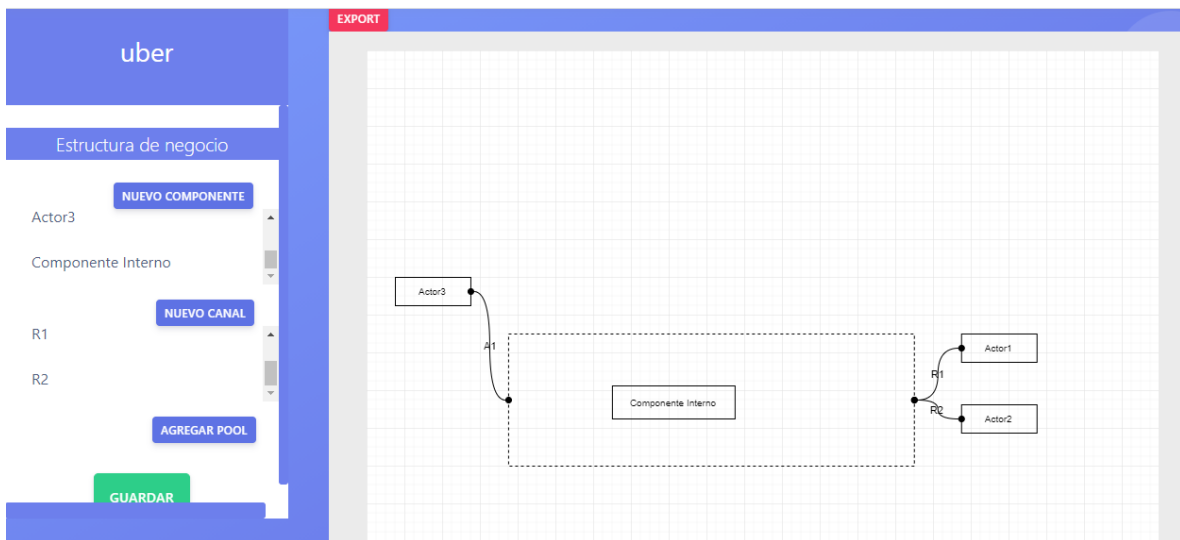


Figura 13. Página con el editor de la estructura del modelo de negocio. Los componentes que representan a los actores son generados automáticamente.

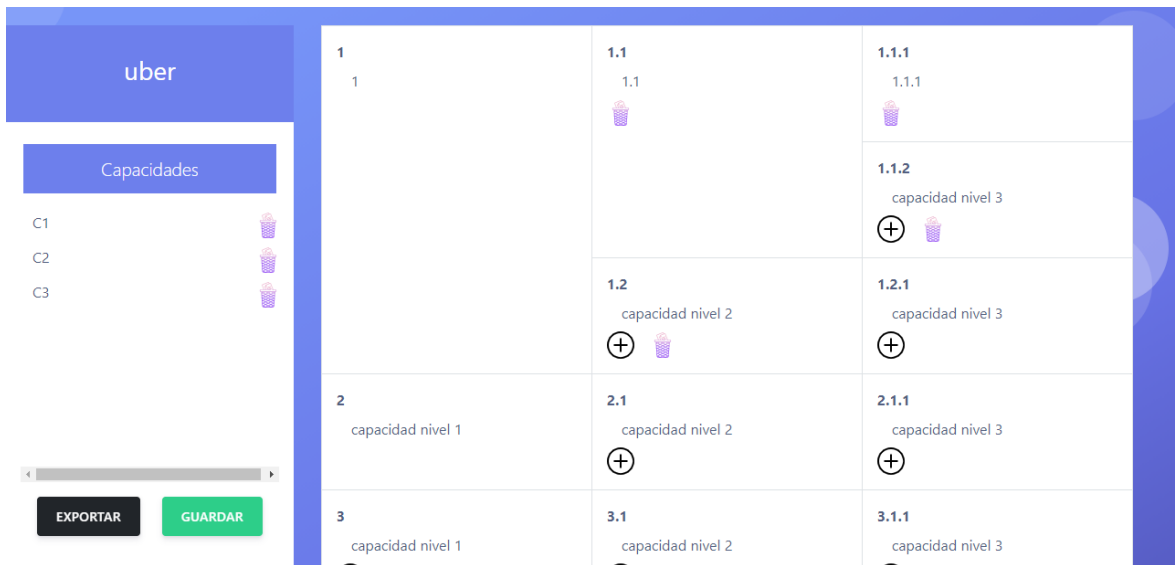


Figura 14. Página con el editor del mapa de capacidades.

Como se puede evidenciar en las anteriores figuras, el entregable final cumple con su objetivo de permitir la realización de los componentes del modelo de negocio.

### 5.1 Descripción de la Implementación

En cuanto a la implementación, se decidió utilizar metodologías ágiles dentro del proyecto debido a que el proyecto debía ser iterativo y adaptarse rápidamente a la retroalimentación y a los cambios. Debido a esto, el proyecto se dividió en 4 Sprints, con sus respectivos entregables:

- Sprint 1: Diseño (agosto 10 – septiembre 7): Durante este sprint se realizaron los pasos de diseño expuestos anteriormente y se estableció una versión esqueleto de la solución mediante un landing page.
- Sprint 2: Implementación base (septiembre 7 – septiembre 28): Durante este sprint se realizaron las funcionalidades de crear, editar y eliminar proyectos. Además, se desarrollo el primer diagrama editable y se incluyó la librería gráfica en el proyecto.
- Sprint 3: Creación de modelos (octubre 5 – octubre 26): Durante este sprint se realizaron la mayoría de los modelos, incluyendo la estructura del modelo de negocio. El entregable de este sprint era la aplicación web con un modelo ontológico y un modelo de canales funcionales.
- Sprint 4: Ajustes y despliegue (octubre 26 – noviembre 30): En este sprint se revisaron los modelos restantes y se realizó la inclusión de tablas para algunos modelos como el portafolio de servicios y el mapa de capacidades. Además, se añadieron algunas funcionalidades para mejorar la experiencia de usuario como ajustar los menús y utilizar un lenguaje coherente. Por último, se realizó el despliegue del proyecto y se hizo el cierre.

Si bien estos requerimientos fueron definidos desde un comienzo, el proyecto fue lo suficientemente flexible para realizar ajustes al cronograma y editar algunos requerimientos para cada sprint a medida que avanzaba el desarrollo.

En las siguientes figuras se presentan los cronogramas del proyecto.

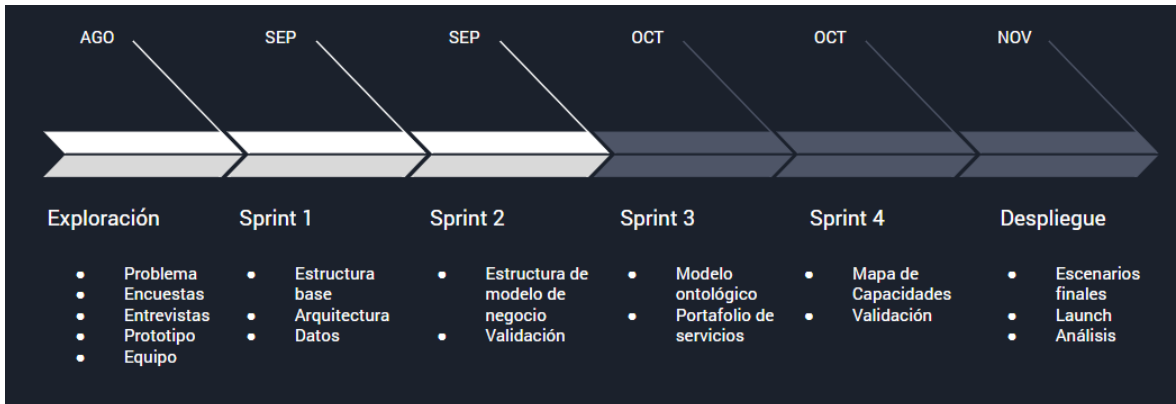


Figura 15. Cronograma Inicial del Proyecto.

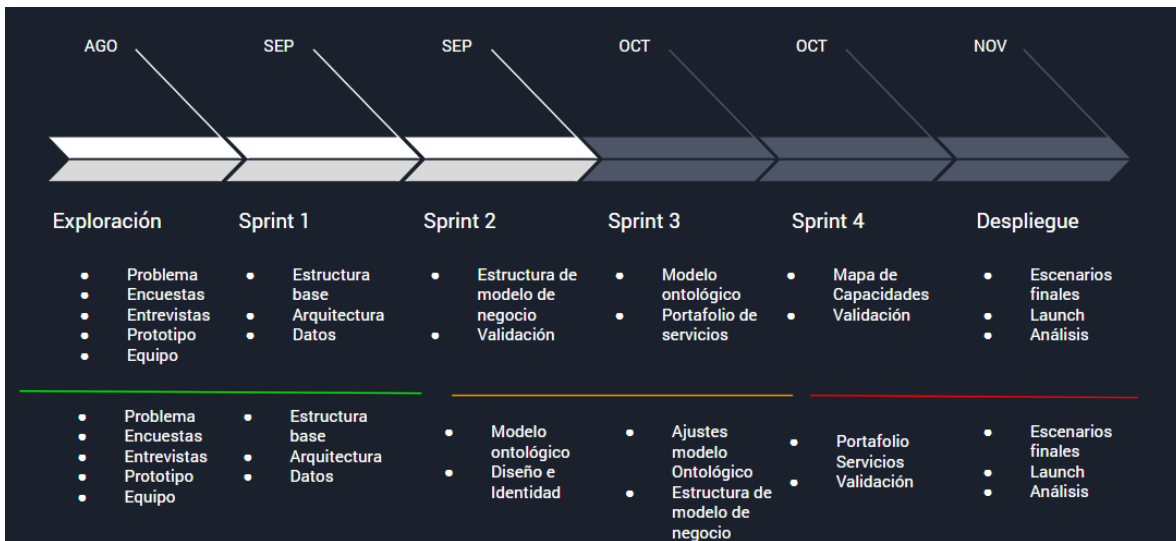


Figura 16. Estado y ajustes del cronograma el 19 de octubre de 2020

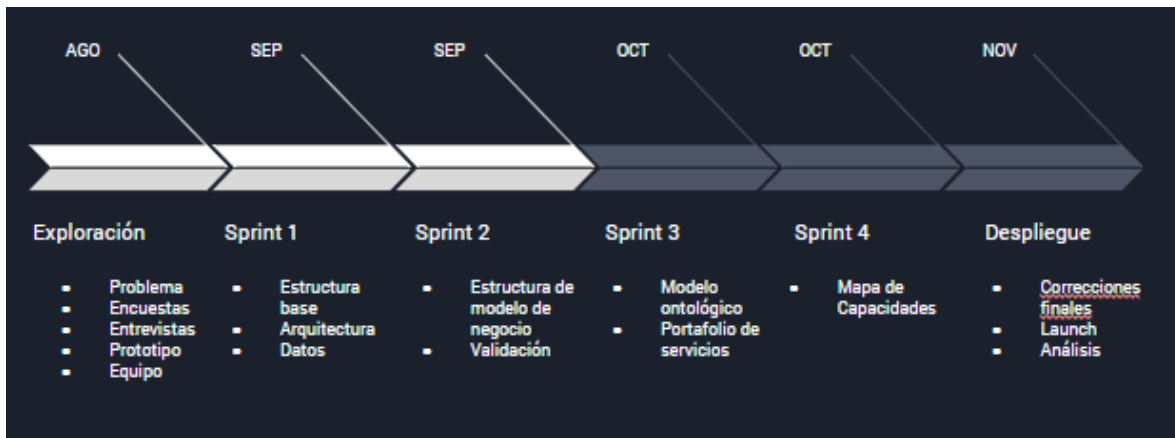


Figura 17. Cronograma Final del proyecto

Cabe resaltar que, si bien hubo ajustes al cronograma, todos los componentes definidos inicialmente se realizaron.

Al utilizar metodologías ágiles, también se aplicaron algunos frameworks que pertenecen a esta metodología. El primer framework utilizado es SCRUM, para el cual el Scrum Master evalúa y da recomendaciones a los otros miembros del equipo de desarrollo. Otro componente de este framework es la utilización de reuniones de corto tiempo constantes, en la que se revisa lo que se ha realizado y lo que falta por realizar y se resuelven dudas. Además, se utilizó el framework de Kanban para realizar tableros con las tareas y de esta manera, organizar mejor el trabajo. En la siguiente figura se puede evidenciar el tablero utilizado mediante la herramienta Trello.

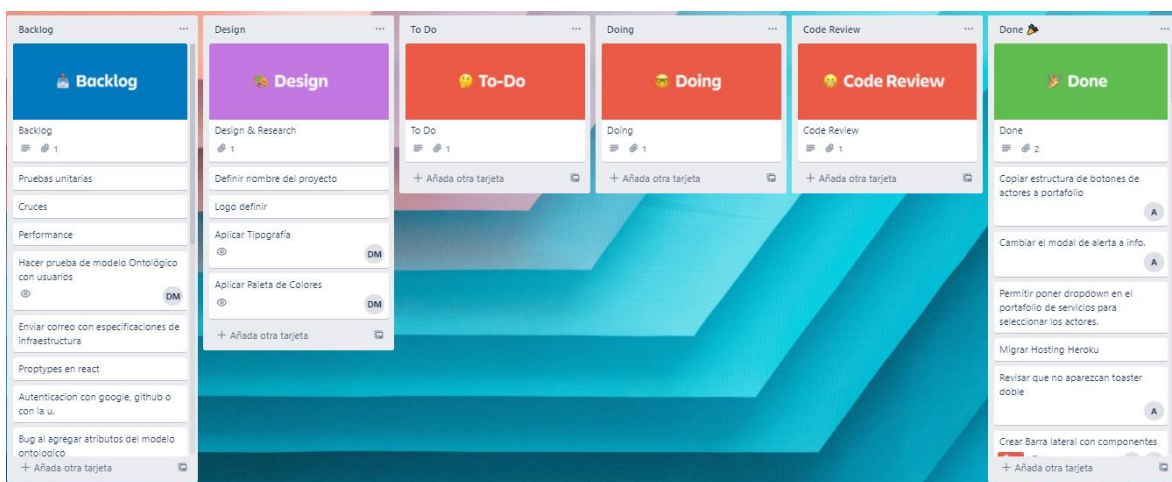


Figura 18. Tablero utilizado para realizar seguimiento a las tareas definidas.

## 5.2 Resultados esperados

Los avances del proyecto se fueron manejando en un repositorio central mediante el uso de técnicas de integración continua. Cada desarrollador tenía su propia rama y cuando un requerimiento estaba completo, actualizaban la rama principal. De esta manera, se garantizaba que el código en la rama principal siempre fuera de calidad. Además, gracias al manejo de las tareas mediante un Kanban, siempre había un seguimiento a los resultados obtenidos semana a semana.

En cuanto a la solución final propuesta, funcionalmente cumple con los objetivos planteados y con los requerimientos funcionales que se definieron. A pesar de eso, puede existir la posibilidad de mejora en la experiencia del usuario. Comparado con otras herramientas existentes el lenguaje utilizado es más simple y se basa en pocos conceptos para realizar cada modelo.

## 6 VALIDACIÓN

### 6.1 Métodos

Con el fin de realizar las pruebas de la aplicación, se volvió a los requerimientos funcionales y se trató de realizar cada una de las funcionalidades. Además, se revisó la interacción de los componentes dentro de cada uno de los modelos con el fin de verificar de que todos los modelos que se pudieran crear fueran sintácticamente correctos. Finalmente, se detectaron ciertos bugs de la aplicación durante la interacción, los más críticos fueron corregidos y los de menor impacto fueron documentados en un backlog de issues.

### 6.2 Validación de Resultados

Para realizar la validación de la aplicación se le presentó la página web a estudiantes y a consultores con el fin de que interactuaran con los diversos elementos y así obtener su retroalimentación. Cabe resaltar que, debido a la naturaleza iterativa del proyecto, la validación no solamente se realizó al final del proyecto, sino que constantemente se presentaba a los distintos usuarios para obtener su retroalimentación. La opinión de los usuarios fue en general positiva y sus recomendaciones fueron solucionadas o asignadas como issues a futuro.

## 7 CONCLUSIONES

### 7.1 Discusión

En conclusión, durante este trabajo se evidenció el manejo de un proyecto de desarrollo de software desde su etapa de diseño hasta la etapa de despliegue. Durante el proceso las metodologías ágiles permitieron un mejor seguimiento y desarrollo del producto y garantizaron una entrega de calidad. Se espera que la solución pueda ser utilizada en ambientes académicos y de consultoría para

representar negocios fácilmente. Por último, esta herramienta puede ser extendida para mejorar la experiencia del usuario final y sus funcionalidades.

## 7.2 Trabajo a Futuro

Si bien la aplicación entregada permite realizar todos los componentes del modelo de negocio de manera adecuada, existen posibilidades de mejora, e incluso de extensión para el proyecto. A continuación, se presentan las principales funcionalidades que podrías ser trabajadas a futuro.

- Realización de cruces entre modelos: Actualmente, el catálogo de actores permite reutilizar sus elementos para realizar el portafolio de servicios y la estructura del modelo de negocio. Sin embargo, se podrían generar nuevas vistas usando los elementos de los modelos que ya existen. Por ejemplo, se podría realizar un cruce de canales con operaciones del portafolio de servicios o una vista con los escenarios de negocio en el modelo de canales.
- Implementación de ejemplos: En la actualidad los modelos comienzan en blanco y los usuarios tienen la libertad de crear sus proyectos y editarlos. A pesar de eso, sería conveniente subir algunos ejemplos reales de proyectos que puedan ser accedidos globalmente por los usuarios en una pestaña de ejemplos.
- Desarrollo de Templates: Aunque mediante esta implementación se eliminó la necesidad de crear las tablas a mano al igual que los componentes gráficos, el usuario podría comenzar mediante plantillas prediseñadas dependiendo de la industria.
- Diseño de pruebas unitarias: Si bien se pueden evidenciar las funcionalidades, es necesario a futuro plantear pruebas unitarias que validen el uso correcto de los componentes de la aplicación en diversos escenarios.

## 8 REFERENCIAS

Moody, Daniel (2009). The “Physics” of notations: Towards a Scientific Basis to Constructing Visual Notations in Software Engineering. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 35, NO. 6, NOVEMBER/DECEMBER 2009

Moody et al (2003). Evaluating the Quality of Information Models: Empirical Testing of a Conceptual Model Quality Framework. Proceedings of the 25<sup>th</sup> International Conference on software Engineering (ICSE '03). The Computer Society.

Villalobos, Jorge. (2020). Modelos y Metamodelos - Lecturas para Arquitectos de Negocio.

Villalobos, Jorge. (2019). El Modelo de Negocio - Lecturas para Arquitectos de Negocio.

Villalobos, Jorge. (2019). El Mapa de Capacidades de Negocio - Lecturas para Arquitectos de Negocio.

Villalobos, Jorge. (2019). Presentaciones de clase. Arquitectura Empresarial 2020-1. Sesiones 5,6,7 y 10.

## BIBLIOGRAFÍA

- Archi (2020). Archi – Open Source Archimate Modelling. Recuperado de <https://www.archimatetool.com/>
- Bizagi (2020) Bizagi. Recuperado de <https://www.bizagi.com/>
- Lucidchart (2020). Diagramación Inteligente. Recuperado de <https://www.lucidchart.com/pages/es>
- Draw.io (2020). Jgraph/drawio. Recuperado de <https://github.com/jgraph/drawio>
- mxGraph (2020). mxGraph 4.2.2. Recuperado de <https://jgraph.github.io/mxgraph/>
- mxGraph (2020). Jgraph/mxgraph. Recuperado de <https://github.com/jgraph/mxgraph>

## **ANEXOS**

Anexo 1. URL a la aplicación final:

<https://arquiwork.herokuapp.com/>

Anexo 2: Repositorio del proyecto:

<https://github.com/theluckyauthor/ArquiNegocio>

Anexo 3. Modelo de datos:

<https://lucid.app/lucidchart/invitations/accept/2acdd436-e875-4dff-aa7b-8b1198e8abad>

Anexo 4: Resultados de la encuesta:

[https://docs.google.com/spreadsheets/d/1LW2aeJCOy46fwrr14\\_ttfpx9Cvdi9\\_JheL4cGPY4kA/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1LW2aeJCOy46fwrr14_ttfpx9Cvdi9_JheL4cGPY4kA/edit?usp=sharing)

Anexo 5: Paleta de Colores:

<https://drive.google.com/file/d/1smWQy8y0ISDV5LIV28fxRj2CN5SeqY1D/view?usp=sharing>

Anexo 6: Propuesta de Diseño:

[https://drive.google.com/file/d/1biUFiq3\\_xHOjTSfkA7yLYIHBtBxUdJyS/view?usp=sharing](https://drive.google.com/file/d/1biUFiq3_xHOjTSfkA7yLYIHBtBxUdJyS/view?usp=sharing)

Anexo 7: Prototipo inicial:

<https://www.figma.com/file/fPVktlnnkRielx57Sm81uV/Prototipo-ArquiNegocio?node-id=6%3A2>

Anexo 8: Mockup de la aplicación:

<https://www.figma.com/file/OIC1in5dShhOML6ir6daTw/Mockup-arquinegocio?node-id=0%3A1>