

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**DESARROLLO DE UN SISTEMA DE SALA DE REUNIONES
PARA LA DEFINICIÓN DEL PERÍMETRO DE BUSQUEDA DE
INFORMACIÓN EN VIGILANCIA ESTRATÉGICA**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

JOSUÉ DANIEL DÍAZ VELOZ

josue.diaz@epn.edu.ec

DIRECTOR: PhD EDISON LOZA AGUIRRE

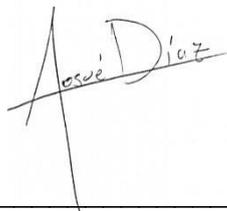
edison.loza@epn.edu.ec

Quito, mayo de 2021

DECLARACION

Yo, Josué Daniel Díaz Veloz declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedo mi derecho de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

A handwritten signature in black ink that reads "Josué Daniel Díaz Veloz". The signature is written in a cursive style and is positioned above a horizontal line.

Josué Daniel Díaz Veloz

CERTIFICACION

Certifico que el presente trabajo fue desarrollado por Josué Daniel Díaz Veloz, bajo mi supervisión.

A handwritten signature in purple ink, reading "Edison Loza", with a long horizontal stroke extending to the right.

PhD. EDISON LOZA AGUIRRE
DIRECTOR DE PROYECTO

AGRADECIMIENTOS

Primero me gustaría agradecer a mi familia por estar conmigo en todas las etapas del proceso de la carrera y por su apoyo incondicional. Gracias por creer en mí y darme una carrera para mi futuro. Me gustaría agradecer a mi tutor académico el PHD Edison Loza Aguirre por todo el apoyo y consejo en la elaboración de este proyecto sin su ayuda todo este proyecto hubiera sido muy complicado para lograr y por eso estaré eternamente agradecido con usted. Por último, agradecer a mi compañero Andrés Torres que estábamos elaborando nuestras tesis al mismo tiempo y nos brindamos apoyo y ánimos a cada paso cuando sentíamos que no estabas avanzando. De nuevo muchas gracias a todos que han estado en este camino, no lo podría haber logrado si me hubiera faltado uno de ustedes, muchas gracias.

DEDICATORIA

Me gustaría dedicar con mucho cariño a mis padres que me apoyaron por cada paso de esta carrera y por fin podrán ver la culminación de todos sus esfuerzos en mi vida. Gracias por darme la oportunidad de conseguir una carrera para mi futuro. También dedicarle a mi hermano y hermana que siempre estuvieron ahí para darme ánimos cuando parecía que perdía todas las fuerzas y ánimos. Gracias a todos.

CONTENIDO

RESUMEN.....	10
ABSTRACT.....	11
1 MARCO TEÓRICO	12
1.1 Introducción	12
1.2 Objetivos	13
1.2.1 General.....	13
1.2.2 Específicos	13
1.3 Alcance	13
1.4 Revisión de Literatura	14
1.4.1 Vigilancia Estratégica.....	14
1.4.2 Targeting.....	16
1.5 Herramientas de Desarrollo	20
1.5.1 Back-End	21
1.5.2 Front-End.....	23
2 METODOLOGÍA.....	26
2.1 Metodología del Prototipado	26
2.2 Aplicación de la Metodología	30
2.2.1 Prototipo de Baja Fidelidad.....	30
2.2.1.1 Plan Rápido	30
2.2.1.2 Desarrollo del Prototipo	31
2.2.1.3 Evaluación del prototipo	36
2.2.2 Prototipo de Media Fidelidad	36
2.2.2.1 Plan Rápido	36
2.2.2.2 Desarrollo del Prototipo	38
2.2.2.3 Evaluación del Prototipo	42
2.2.3 Prototipo de Alta Fidelidad.....	42
2.2.3.1 Plan Rápido	42
2.2.3.2 Desarrollo del Prototipo	43
2.2.3.3 Evaluación del Prototipo	46

2.3	Metodología de Evaluación	46
3	RESULTADOS	48
3.1	Descripción del Sistema.....	48
3.1.1	Descripción de Arquitectura.....	48
3.1.2	Descripción de la Aplicación Web.....	53
3.2	Resultados de la Evaluación de Aceptación	68
4	CONCLUSIONES.....	71
5	RECOMENDACIONES	72
BIBLIOGRAFÍA.....		73
GLOSARIO		76

ÍNDICE DE TABLAS

Tabla 1. Ejemplo de la matriz de targeting. Elaborado por autor.....	17
Tabla 2. Herramientas de Back-End. Elaborada por el autor.	22
Tabla 3. Herramientas de Front-End. Elaborada por el autor.....	25
Tabla 4. Listado de Historias de Usuarios para el prototipo de baja fidelidad. Elaborado por autor.	31
Tabla 5. Listado de Historias de Usuarios para el prototipo de media fidelidad. Elaborado por autor.	37
Tabla 6. Listado de Historias de Usuarios para el prototipo de alta fidelidad después de observaciones. Elaborado por autor.	42
Tabla 7. Listado de Historias de Usuarios para el prototipo de alta fidelidad. Elaborado por autor.	43

ÍNDICE DE FIGURAS

Figura 1. Proceso de SScan [8].	15
Figura 2. Ejemplo matriz resultante de targeting [2].	18
Figura 3. Método de Target [2].	18
Figura 4. Ciclo de Vida del Software basado en Prototipado [35].	27
Figura 5. Ciclo de refinamiento de prototipo [39].	28
Figura 6. Vista general de las pantallas en AdobeXD. Elaborado por el autor.	32
Figura 7. Pantalla de Inicio de Sesión - Prototipo Baja Fidelidad. Elaborado por el autor.	33
Figura 8. Pantalla de Registro de Usuario - Prototipo Baja Fidelidad. Elaborado por el autor.	33
Figura 9. Pantalla de Inicio - Prototipo Baja Fidelidad. Elaborado por el autor.	34
Figura 10. Pantalla de Actores - Prototipo Baja Fidelidad. Elaborado por el autor.	35
Figura 11. Pantalla de Temas - Prototipo Baja Fidelidad. Elaborado por el autor.	35
Figura 12. Pantalla de Registro de Matriz - Prototipo Baja Fidelidad. Elaborado por el autor.	36
Figura 13. Repositorio creado en GitHub para el desarrollo del prototipo de media fidelidad. Elaborado por el autor.	38
Figura 14. Pantalla de Inicio de Sesión. Elaborada por el autor.	39
Figura 15. Pantalla de Crear Usuario. Elaborada por el autor.	39
Figura 16. Pantalla de Inicio. Elaborada por el autor.	40
Figura 17. Pantalla de Actores. Elaborada por el autor.	40
Figura 18. Pantalla de Temas. Elaborada por el autor.	41
Figura 19. Pantalla de Matriz. Elaborada por el autor.	41
Figura 20. Pantalla de Actores. Elaborada por el autor.	44
Figura 21. Pantalla de Temas. Elaborada por el autor.	44
Figura 22. Pantalla de Matrices. Elaborada por el autor.	45
Figura 23. Pantalla de Matriz. Elaborada por el autor.	46
Figura 24. Modelo TAM [41].	47
Figura 25. Modelo lógico de la base de datos. Elaborado por el autor.	48
Figura 26. Modelo de datos desplegado con Prisma. Elaborado por el autor.	49
Figura 27. Diagrama de la arquitectura de la aplicación web. Elaborado por el autor.	50
Figura 28. Protocolo de WebSocket. Elaborado por el autor.	51
Figura 29. Link de WebSocket. Elaborado por el autor.	52
Figura 30. Ejemplo de un subscriptor. Elaborador por el autor.	53
Figura 31. Pantalla inicial de la aplicación web. Elaborado por el autor.	53
Figura 32. Página de Registro de Usuario de la aplicación web. Elaborado por el autor.	54
Figura 33. Página Principal de la aplicación web. Elaborado por el autor.	54
Figura 34. Página de Actores de la aplicación web. Elaborado por el autor.	55
Figura 35. Formulario de Creación de Actor tipo Padre de la aplicación web. Elaborado por el autor.	55

Figura 36. Comprobación de la creación de un actor tipo padre. Elaborado por el autor.	56
Figura 37. Formulario de Creación de Actor tipo sub-actor de la aplicación web. Elaborado por el autor.	56
Figura 38. Comprobación de la creación de un actor tipo sub-actor. Elaborado por el autor.	57
Figura 39. Ejemplo de una lista completa de actores. Elaborada por el autor.	57
Figura 40. Mensaje de confirmación para eliminar actor de la aplicación web. Elaborado por el autor.	58
Figura 41. Formulario de edición de un actor de la aplicación web. Elaborado por el autor.	58
Figura 42. Ejemplo del filtro por actor de la aplicación web. Elaborado por el autor.	59
Figura 43. Página de Temas de la aplicación web. Elaborado por el autor.	59
Figura 44. Formulario de Creación de Tema tipo Padre de la aplicación web. Elaborado por el autor.	60
Figura 45. Comprobación de la creación de un tema tipo padre. Elaborado por el autor.	60
Figura 46. Formulario de Creación de Tema tipo sub-tema de la aplicación web. Elaborado por el autor.	61
Figura 47. Comprobación de la creación de un actor tipo sub-tema. Elaborado por el autor.	61
Figura 48. Ejemplo de una lista completa de temas. Elaborada por el autor. ..	62
Figura 49. Mensaje de confirmación para eliminar tema de la aplicación web. Elaborado por el autor.	62
Figura 50. Formulario de edición de un tema de la aplicación web. Elaborado por el autor.	63
Figura 51. Ejemplo del filtro por tema de la aplicación web. Elaborado por el autor.	63
Figura 52. Página de Matrices de la aplicación web. Elaborado por el autor. ..	64
Figura 53. Mensaje de confirmación de creación de matriz de la aplicación web. Elaborado por el autor.	64
Figura 54. Mensaje de confirmación para eliminar matriz de la aplicación web. Elaborado por el autor.	65
Figura 55. Página de Matriz de la aplicación web. Elaborado por el autor.	66
Figura 56. Filtro de Actor por prioridad de la aplicación web. Elaborado por el autor.	66
Figura 57. Filtro de Tema por prioridad de la aplicación web. Elaborado por el autor.	67
Figura 58. Filtro por prioridad de actor y tema en la aplicación web. Elaborado por el autor.	67
Figura 59. Ejemplo de una matriz de targeting completa. Elaborado por el autor.	68
Figura 60. Modelo de encuesta. Elaborado por el autor.	68
Figura 61. Resultados facilidad de uso del sistema. Elaborado por el autor. ...	69
Figura 62. Resultados utilidad percibida del sistema. Elaborado por el autor. .	70

RESUMEN

La vigilancia estratégica es el proceso que permite a las organizaciones estar al tanto de los cambios que ocurren en su entorno, con el objetivo de tomar las decisiones más adecuadas para preservar las ventajas competitivas que les permitan asegurar su posición futura en el mercado. Este proceso, a pesar de ser muy importante y de gran ayuda para las empresas, carece de herramientas tecnológicas que asistan a los participantes en las diferentes etapas del proceso, forzando en muchos casos al uso de herramientas físicas y tradicionales como pizarrones y marcadores. Con esto en mente, el presente proyecto aborda este problema al proponer una herramienta que asista a la etapa de targeting del proceso de la vigilancia estratégica.

El targeting es la etapa del proceso que delimita el alcance de búsqueda de la vigilancia estratégica. Este alcance debe estar alineado con los objetivos estratégicos y prioridades de la empresa. El proceso de targeting toma las necesidades de información de una empresa y las representa en una relación de actores y temas. Este método también toma la interrelación creada para ayudar a eliminar el escaneo aislado de cada actor o tema. El sistema propuesto en esta tesis se enfoca en facilitar y optimizar este proceso usando un sistema tipo “meeting room” que brindará una interacción en tiempo real a los usuarios así creando un ambiente colaborativo. Se optó por este tipo de sistema ya que el proceso de targeting es un proceso colaborativo que se lleva a cabo con varios miembros de las juntas directivas para poder analizar la información propuesta y tomar una decisión basada en ella.

El sistema desarrollado fue evaluado mediante experimentaciones en base a los dos criterios predominantes en los modelos de evaluación de aceptabilidad de tecnología. Sobre esta evaluación se obtuvo que un 88% de las personas consideraron que la aplicación ofrece facilidad de uso y un 100% en su utilidad percibida para el proceso de targeting, concluyendo que, a pesar de necesitar pequeñas mejoras en su facilidad de uso, el sistema permite de manera decisiva apoyar y optimizar los procesos de vigilancia estratégica.

ABSTRACT

Strategic scanning is the process that allows organizations to be aware of the changes that occur in their environment, with the objective of making the most appropriate decisions to preserve their competitive advantages. Even when this process is very important and helps companies, currently there is a lack of technological tools to assist participants in the different stages of the process, therefore traditional tools, such as blackboards and markers, are used. With this in mind, this project deals with this problem in the Targeting stage of the strategic scanning process.

Targeting is the stage of the process that defines the scope of the search. This scope must be aligned with the strategic objectives and priorities of the company. The targeting process takes organizational information needs and represents it in a relationship between actors and themes. This method will use the interrelationship found to help eliminate the isolated scan of each actor or subject. The proposed system will focus on facilitating and optimizing this process using a “meeting room” system that will offer real-time interaction to its users in a collaborative environment. This type of system was chosen since the targeting process is a collaborative process that is carried out by several members of the boards of directors that analyze the proposed information and make a decision based on it.

The developed system was evaluated through experimentation based on the two predominant criteria in the technology acceptance model. On this evaluation, the 88% of the evaluators considered that the application is ease of use and 100% perceived it as usefulness for the targeting process. Thus, despite some small improvements in its ease of use, this system makes it possible to decisively support and optimize the strategic scanning process.

1 MARCO TEÓRICO

1.1 Introducción

La Vigilancia Estratégica (*Strategic Scanning, SScan*) consiste en la recopilación y manejo de información que incluye eventos, tendencias, noticias o datos relacionados con el entorno de una organización; la cual se usa para ayudar a los altos ejecutivos a identificar y entender las amenazas y oportunidades estratégicas que les permitan mejorar la toma de decisiones de la organización [1].

A pesar del beneficio de obtener esta información externa, una organización no dispone de los medios para vigilar todo su entorno socio económico. Así, en SScan, el proceso de definir el perímetro de búsqueda de información relevante para los ejecutivos, basándose en las expectativas y objetivos de la compañía, se llama Targeting [2]. Este proceso puede ser bastante complejo ya que involucra la colaboración de varios gerentes provenientes de varias unidades de la organización que pueden tener diferentes intereses y/o vocabularios.

Aun cuando existen varios sistemas enfocados a la recolección y análisis de información para su uso en SScan (e.g. [3,4,5]), actualmente existen pocas soluciones para ayudar en la fase de Targeting (e.g. [2]). Una alternativa para apoyar a la definición del perímetro en el Targeting son los sistemas tipo sala de reuniones (meeting room systems). Estos sistemas se enfocan en la colaboración que ofrece un entorno informatizado que utiliza equipos disponibles para el uso habitual de los empleados, con el objetivo de facilitar la toma de decisiones colectivas [6]. Algunos ensayos se han desarrollado a nivel de demostrador para utilizar sistemas de sala de reuniones en tareas de Targeting [2]. En este proyecto se propone continuar con la creación de una plataforma web que permita realizar estas tareas, pero tomando ventaja de la interacción de los usuarios a través de sus dispositivos móviles y de pantallas interactivas con el fin de que el proceso de Targeting se más eficaz y amigable.

1.2 Objetivos

1.2.1 General

- Desarrollar un sistema de sala de reuniones para la definición del perímetro de recolección de información en vigilancia estratégica basado en el uso de pantallas interactivas, aplicaciones móviles y aplicaciones web.

1.2.2 Específicos

- Diseñar e implementar una plataforma web interactiva y amigable para los usuarios que forman parte del personal de una organización, que muestren reportes e información de valor para apoyar la toma de decisiones de la organización.
- Incorporar un sistema interactivo y amigable que sea compatible con dispositivos móviles que permita la recolección de información, visualización de datos e información de valor para apoyar la toma de decisiones de la organización.
- Evaluar la aceptabilidad de los sistemas propuestos en términos de utilidad y facilidad de uso percibida mediante experimentaciones.

1.3 Alcance

En el presente proyecto, se desarrolla un sistema de sala de reuniones para la definición del perímetro de recolección de información en el proceso de SScan. El proyecto se enfocará en la fase de Targeting de SScan. El presente proyecto no se enfocará en la recolección de información y data que es parte de las etapas posteriores de SScan. El objetivo de este proyecto es facilitar el proceso de Targeting de una manera interactiva e intuitiva para los usuarios, creando un sistema colaborativo entre ellos. Primordialmente, este proyecto ayudará al proceso de toma de decisiones ya que el sistema facilitará la presentación de información relevante e influyente para la compañía y organización.

1.4 Revisión de Literatura

1.4.1 Vigilancia Estratégica

El SScan es una actividad que ayuda a las compañías, en especial a la alta gerencia, en el proceso de toma de decisiones. SScan se refiere a “la adquisición y uso de información sobre eventos, tendencias y relaciones en el entorno externo de una organización, cuyo conocimiento ayudará a la gerencia a planificar el futuro curso de acción de la organización” [1]. Este puede servir, por ejemplo, para que una organización pueda mantenerse informada de las tendencias y los desarrollos en el mercado [7]. Con el conocimiento de estas tendencias y desarrollos se pueden identificar limitaciones, amenazas y oportunidades reduciendo así el riesgo vinculado a la incertidumbre.

El proceso de SScan, que se puede visualizar en la Figura 1, se puede dividir en dos etapas. La primera, conocida como la etapa de recolección de datos, consiste en la definición del problema que se va a monitorear y la delimitación del perímetro del entorno externo donde se va a recolectar información [7]. El primer paso es encontrar el problema que necesita atención o es crítico para la empresa. Una vez que se encuentra un problema que puede involucrar una decisión estratégica, se procede a la identificación de la parte de su entorno empresarial a ser monitoreado para recolectar información relacionada con el problema. En esta fase es donde se define la información útil contra la inútil, para poder lograr esto se tiene que asegurar que el mecanismo adecuado para la transmisión, organización y filtrado de la información recabada sea puesto en uso [8].

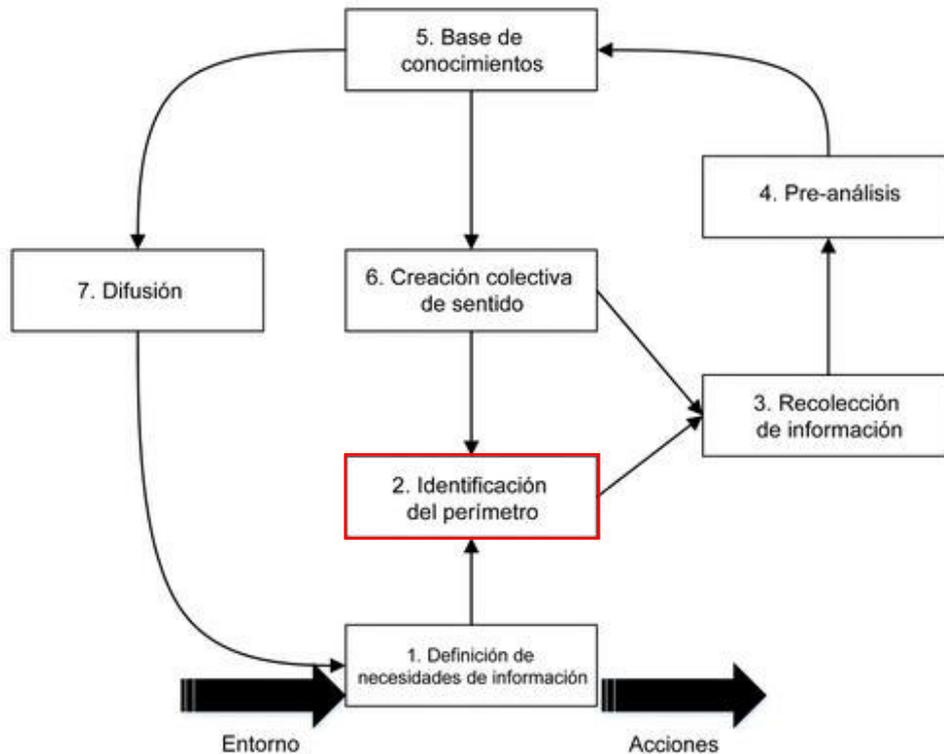


Figura 1. Proceso de SScan [8].

La segunda etapa de SScan es conocida como la etapa de análisis de información. Esta etapa empieza con una fase de preanálisis donde la información se organiza, reduce, enriquece, y se relacionan las diferentes piezas de información colectada con el objetivo de darle sentido [8]. Esto se puede lograr con ayuda de medios humanos y/o tecnológicos. Toda esta información recolectada y procesada es guardada en un tipo de base de datos o base de conocimiento para así seguir al siguiente paso que es la creación colectiva de sentido. Construir colectivamente un sentido consiste en reunir a los miembros de la organización que pueden interpretar la información guardada en la base de conocimientos con el punto de vista organizacional y con el objetivo de tomar decisiones basadas en ella [7]. Es en esta etapa donde se evalúa si la información es suficiente o si es necesario recategorizar, eliminar, añadir o editar la información. Dependiendo de esto se decide si se vuelve a la definición de perímetro o solo a recolectar más datos. Si la información recolectada es suficiente para poder generar un punto de vista de la organización y ayudar a la toma de decisiones se avanza a la etapa de difusión de la información relevante y de las decisiones que se van a tomar.

1.4.2 Targeting

La esencia del SScan es la recolección de información, pero esto puede ser un proceso complicado en la práctica ya que muchas veces la gerencia no sabe que está buscando específicamente, o puede caer en un espectro muy amplio de búsqueda [2]; lo cual resulta en un exceso de información irrelevante que puede conllevar a la pérdida de información importante. La solución a este problema se lo conoce como “Targeting”.

El Targeting se enfoca en la delimitación del alcance de búsqueda de información [2]. Este alcance tiene que estar alineado con los objetivos estratégicos y prioridades de la compañía. En la actualidad existen varios sistemas que ayudan en la recolección y análisis de datos [9], pero ninguno enfocado al Targeting. En consecuencia, las gerencias optan por el uso de herramientas físicas como son marcadores y papel para poder realizar este proceso, lo que consume mucho tiempo y recursos.

El proceso de Targeting consiste en tomar las necesidades de información de una organización y representarlas en una relación entre actores y temas [2]. Este método propone un enfoque holístico identificando e interrelacionando temas y actores relevantes para SScan [10]. Este método también se enfoca en la interrelación entre sí de actores y temas que ayuda a eliminar el escaneo aislado de cada actor o tema [2].

Un actor es definido como una persona física o jurídica cuyas decisiones y acciones podrían influir en el futuro de la organización y sus actividades. Un tema en cambio se refiere a un centro de interés cuando se considera el futuro de la organización [9]. Al definir actores y temas puede ocurrir casos donde un actor no tenga relación con ningún tema o que un actor y un tema que no tengan relación entre sí. Para poder visualizar y definir estas interrelaciones se usa una herramienta llamada “Matriz de Targeting”. Esta matriz es usada por los participantes del proceso de Targeting donde ellos identificarán los cruces entre actores y temas que tienen relevancia e importancia para ellos y para la organización [2], un ejemplo de esta matriz se puede visualizar en la Tabla 1.

	Actor 1	Actor 2	Actor 3
Tema 1		X	X
Tema 2	X	X	
Tema 3			X

Tabla 1. Ejemplo de la matriz de targeting. Elaborado por autor.

La matriz es uno de los resultados del proceso de Targeting, donde en cada celda se puede generar una marca o comentario para definir que ese específico actor y tema tiene una relevancia e interrelación entre sí (una X en la Tabla 1). Así, al terminar el proceso se obtiene una lista nominativa de actores, una lista precisa de temas, y una matriz de Targeting con sus cruces entre actores y temas [2].

En el paper titulado “Getting a Clean Shot on a Blurred Target: Improving Targeting for Strategic Scanning through Action Research in 10 French Organizations” [2] tenemos un ejemplo donde se toma el proceso de Targeting y con integración de la tecnología ayudan a identificar problemas relevantes para cada organización del estudio. Los autores proponen algunas mejoras al proceso de Targeting como: incluir una lista de actores y temas para que sirvan de punto de partida para activar y facilitar la discusión; o priorizar los actores y temas más importantes para escanear y así producir matrices objetivas que sean más pequeñas y manejables [2]. Esta investigación evidenció que, al principio, el proceso de Targeting puede ser abstracto para los gerentes de las empresas, pero con la ayuda de la tecnología y de herramientas visuales, este proceso se hizo mucho más claro y entendible. Se puede visualizar en la Figura 2 los resultados que se obtuvieron. La principal limitación del aporte presentado en este paper reside en que la herramienta desarrollada no permitía interactividad entre los participantes, siendo proyectada en una pantalla durante las reuniones. El presente proyecto pretende sobrepasar esta limitación.

			Customers		Suppliers		
			Competitors	B2B customers	Customers of our customers	Tier 1	Tier 2
Environmental	Products and services	Eco-design	🔴🔴🔴🔴	🟡🟡🟡🟡	🟢🟢🟢🟢	🔴🔴🔴🔴	🔴🔴🔴🔴
Human rights	Investment and procurement practices		🔴🔴🔴🔴	🟡🟡🟡🟡	🟢🟢🟢🟢	🔴🔴🔴🔴	🔴🔴🔴🔴
Logistics and SCM	Forecasts		🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡
	Management of flows		🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡
	Measurement of the logistics performance		🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡
	Mutualisation		🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡
	Sourcing		🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡
Société	Supply chain integration		🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡
	Anti-competitive behavior		🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡
	Corruption		🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡	🟡🟡🟡🟡

Figura 2. Ejemplo matriz resultante de targeting [2].

En este paper se describe el método elaborado y ejecutado para la construcción y obtención de la matriz resultante de la Figura 2. Este método se puede visualizar en la Figura 3.

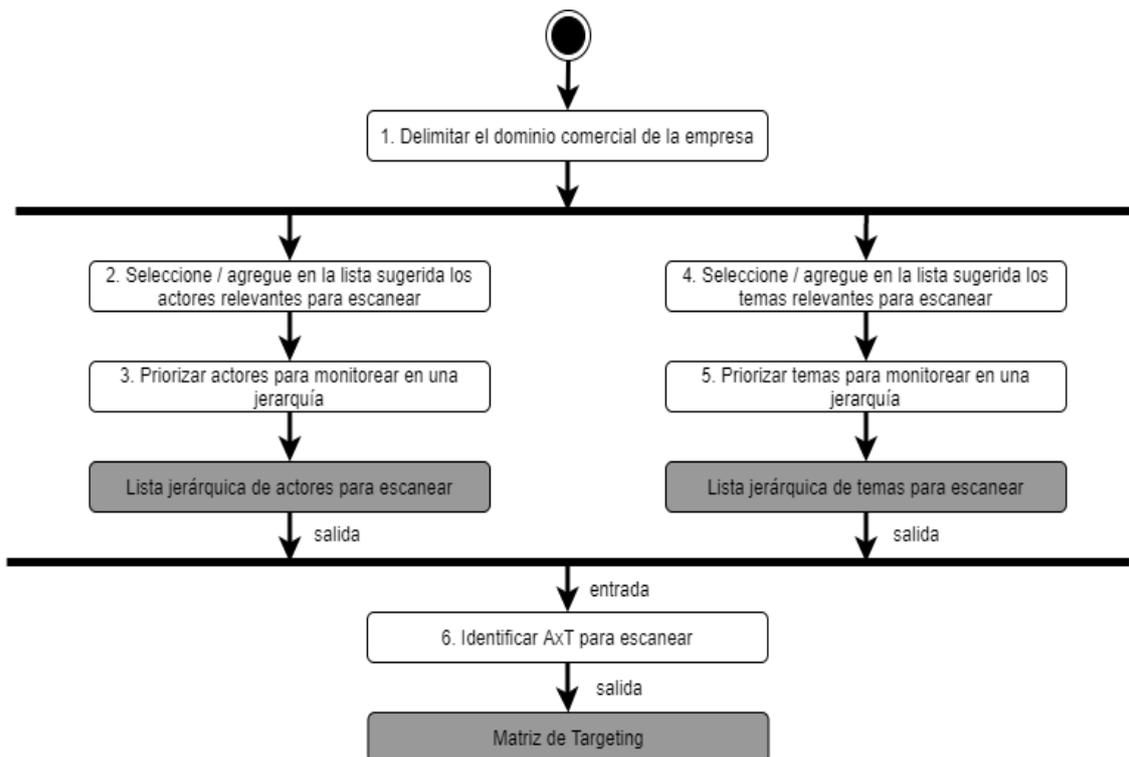


Figura 3. Método de Target [2].

El paso inicial que se necesita para usar el método de target propuesto consiste en delimitar el dominio comercial de la empresa, esto significa, reducir el tipo de información que se está analizando o que puede resultar en dificultades. Estas dificultades pueden llevar a situaciones indeseables como una

sobreabundancia de información irrelevante, que puede agravar el problema de la sobrecarga de datos y convertirse en un estorbo, abrumar a los administradores y hacer que pasen por alto o pierdan información importante y, en consecuencia, paralizar el análisis y la toma de decisiones [2]. Por tanto, es recomendado que se limite la adquisición de información usando los objetivos estratégicos y prioridades de la organización y asegurándose que esta información esté de acuerdo ellos.

El siguiente paso consiste en seleccionar o agregar actores relevantes. Como se mencionó anteriormente el Targeting da como salida el obtener una lista concisa de actores relevantes. Como paso tres se define la prioridad de cada actor, dando como resultado una lista jerárquica, definida por importancia, de actores relevantes.

Este mismo proceso se repite para los temas. Estos también deberán ser definidos con una prioridad para así conseguir la lista jerárquica, definida por importancia, de temas relevantes. Una vez que se tengan estas dos listas, se tiene que identificar los cruces relevantes entre Actores y Temas, o también definidos como cruces AxT. Cada uno de estos cruces AxT serán clasificados también por prioridades para así establecer una jerarquía entre los cruces AxT que se definan y conseguir la matriz de Targeting. Usando este método se ayuda a las empresas, de una manera eficiente y rápida, a conseguir un objetivo claro donde cada empresa deba enfocarse [2].

El presente trabajo pretende proponer el desarrollo de un sistema de tipo meeting room, que se enfoque en la colaboración y ofrezca un entorno automatizado que utiliza equipos disponibles para el uso habitual de los empleados [6]. El sistema estará enfocado en ser una herramienta colaborativa, que sea accesible por sus usuarios, que facilite el proceso de SScan para las compañías y que permita el ahorro de tiempo y recursos a la organización.

1.5 Herramientas de Desarrollo

Con el nacimiento del internet y nuevas tecnologías, el desarrollo web se vuelve crucial para proyectos y negocios alrededor del mundo. El auge de Internet y la economía global e interconectada ha aumentado la necesidad de una adaptación rápida a las necesidades cambiantes del cliente [11]. Hoy en día muchas páginas web y compañías se enfrentan a nuevos problemas como multiusuarios, grandes cantidades de solicitudes, diseños accesibles, y muchas más. Para enfrentar estos problemas es necesario entender el desarrollo web como tal, que se puede definir como un conjunto de métodos, técnicas y herramientas en Ingeniería de Software que ayuda a un equipo de desarrollo a construir sistemas en la Web [11].

Un sistema Web puede dividirse en dos partes: el Back-end (la arquitectura y aplicaciones del lado del servidor) y el Front-end (la arquitectura y aplicaciones del lado del Cliente). El Front-end es la parte que el usuario ve e interactúa, como menús, formularios de contacto, pantallas, iconos etc. Para diseñar y desarrollar dicha interfaz web, se deben utilizar ciertas herramientas y tecnologías, que generalmente son una combinación de HTML, CSS y JavaScript, todo controlado por el navegador [12].

El back-end es esencialmente el componente de un sitio web que el usuario no ve. Es responsable de almacenar y organizar los datos, y de garantizar que todo en el lado del cliente funcione sin problemas. El back-end generalmente consiste de tres partes: un servidor, una aplicación y una base de datos [12]. Entre estos dos componentes existe una comunicación bidireccional donde se transmiten los datos, tomando los datos almacenados en el back-end para así desplegarlos en el front-end para el uso y modificación por parte del usuario, para después ser enviado de vuelta al back-end para almacenar nueva información o actualizar la información almacenada.

1.5.1 Back-End

En el presente proyecto, para el Back-End, se utilizaron las herramientas que se detalla en la Tabla 2.

Nombre	Descripción	Utilizado en
 <p data-bbox="331 645 408 674">Apollo</p>	<p data-bbox="533 465 1106 645">La plataforma Apollo está diseñada para ayudar a construir, consultar y administrar un grafo de datos: una capa de datos unificada que permite a las aplicaciones interactuar con los datos de cualquier combinación de almacenes de datos conectados y con APIs externas.</p> <p data-bbox="533 656 1106 801">El grafo de datos, creado en GraphQL, se encuentra entre los clientes de la aplicación y los servicios de back-end. Apollo facilita el flujo de datos entre el cliente, el grafo de datos, y el servidor [13].</p> <p data-bbox="533 813 1106 992">El Cliente de Apollo es una biblioteca completa de administración de estado para aplicaciones JavaScript. Usa consulta GraphQL para comunicarse con el cliente de Apollo donde este se encargará de solicitar y almacenar en caché los datos [14].</p>	<p data-bbox="1163 465 1332 555">Conexión de Back-End con Front-End</p>
 <p data-bbox="325 1115 411 1144">Docker</p>	<p data-bbox="533 1008 1106 1310">Docker es una plataforma abierta para desarrollar, enviar y ejecutar aplicaciones. Docker permite separar las aplicaciones de su infraestructura para entregar software rápidamente. Con Docker, se puede administrar la infraestructura de la misma manera que se administran sus aplicaciones. Además, ofrece la capacidad de empaquetar y ejecutar una aplicación en un entorno aislado llamado contenedor.</p> <p data-bbox="533 1321 1106 1400">El aislamiento y la seguridad de Docker le permiten ejecutar muchos contenedores simultáneamente en un host determinado.</p> <p data-bbox="533 1411 1106 1532">Docker proporciona herramientas y una plataforma para administrar el ciclo de vida de sus contenedores que incluye desarrollo, pruebas y despliegue [15].</p>	<p data-bbox="1155 1008 1340 1128">Contenedor del servidor GraphQL y la base de datos</p>
 <p data-bbox="300 1666 440 1742">GraphQL GraphQL</p>	<p data-bbox="533 1550 1106 1883">GraphQL es un lenguaje de consulta declarativo para API's y un entorno de ejecución que permite la entrega de resultados para esas consultas. GraphQL provee una completa y entendible descripción de la data en el API, dando a los clientes el poder para pedir exactamente los datos que estos necesitan y nada más, haciendo más fácil para las API's evolucionar en el tiempo, al mismo tiempo que se integra con poderosas herramientas de desarrollo.</p> <p data-bbox="533 1895 1106 2007">GraphQL no está vinculado a ninguna base de datos o motor de almacenamiento específico, sino que está respaldado por su código y datos existentes [16]. Un servicio GraphQL se crea</p>	<p data-bbox="1155 1550 1337 1579">Aplicación web</p>

	<p>definiendo tipos y campos, luego proporcionando funciones para cada campo en cada tipo. Esto ayuda a relacionar las estructuras tradicionales de base de datos a la estructura de GraphQL. Los tipos serían el equivalente de las entidades en una estructura tradicional.</p> <p>Una vez que se ejecuta un servicio GraphQL, este puede recibir consultas GraphQL para validar y ejecutar. Una consulta recibida se verifica primero para asegurarse de que solo se refiere a los tipos y campos definidos, luego ejecuta las funciones proporcionadas para producir un resultado.</p> <p>La ventaja de usar GraphQL es que se pueden definir los datos que se van a recuperar en una estructura de objeto el cual va a facilitar y agilizar el proceso de obtención de datos hecha por el sistema.</p>	
 <p>Lucidchart</p>	<p>Lucidchart es un espacio visual que ayuda a los usuarios a bosquejar y compartir diagramas con sus colaboradores. Lucidchart tiene la función para realizar un trabajo en tiempo real con los colaboradores, combinando diagramas, y visualización de datos [17].</p> <p>Lucidchart tiene varias plantillas y tipos de diagramas para facilitar la creación de diagramas para diferentes situaciones. Algunos tipos de diagramas que ofrece LucidChart son diagramas de flujo, diagramas de base de datos, diagramas de red, mapas conceptuales y muchos diagramas más que pueden ser usados en diferentes entornos como son el desarrollo de software, gestión de proyectos, y muchos otros [18].</p>	<p>Diagramación del modelo de base de datos</p>
 <p>PostgreSQL</p>	<p>PostgreSQL es un sistema de gestión de bases de datos relacionales de objetos basado en POSTGRES. PostgreSQL es un gestor de bases de datos de código abierto que utiliza y amplía el lenguaje SQL, combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas [19].</p>	<p>Base de datos</p>
 <p>Prisma</p>	<p>Prisma es un kit de herramientas de base de datos de código abierto. Prisma Client se puede usar en cualquier aplicación de back-end Node.js o TypeScript. Puede ser una API REST, una API GraphQL, una API gRPC o cualquier otra cosa que necesite una base de datos. Es una herramienta que pretende reemplazar las ORM's (Object Relational Mapping) tradicionales y simplificar flujos de trabajo sobre bases de datos, usando principalmente un cliente autogenerado que soporta seguridad de tipos [20].</p>	<p>Despliegue de la base de datos</p>

Tabla 2. Herramientas de Back-End. Elaborada por el autor.

1.5.2 Front-End

En el presente proyecto, para el Front-End, se utilizaron las herramientas que se detallan en la Tabla 3.

Nombre	Descripción	Utilizado en
 Adobe XD	Es una herramienta diseñada por Adobe enfocada en el proceso de diseño de experiencias e interfaces de usuarios. Adobe XD incluye el trabajo colaborativo con usuarios a través de tiempo real. Está enfocada en crear prototipos navegables que permitan simular la navegabilidad de la página web o aplicación móvil que se ha diseñado, de una manera rápida y eficiente, gracias a las plantillas y herramientas que provee. Adobe XD logra esto por su funciones y características como auto animaciones, vistas previas, interacciones entre pantallas, y simuladores web y móviles para los proyectos [21].	Diseño de Mockups navegables
 Angular	Angular es una plataforma y un framework para crear aplicaciones utilizando HTML y TypeScript. Está escrito en TypeScript. Implementa funcionalidad básica y opcional como un conjunto de bibliotecas TypeScript que se importa a las aplicaciones. Los componentes básicos son NgModules. NgModules recopila código relacionado en conjuntos funcionales. Así, una aplicación angular se define mediante un conjunto de NgModules [22]. Una aplicación siempre tiene al menos un módulo raíz que permite el arranque y, por lo general, tiene muchos más módulos de características. Se define un componente como vistas cuando este conlleva un conjunto de elementos de pantalla y se define como servicios a los que dan un uso funcional y específico que pueden no estar relacionados directamente a las vistas. Usando una mezcla de estos componentes, Angular permite crear navegaciones entre vistas y permite crear la lógica detrás de cada componente para crear aplicaciones web.	Aplicación web
 Angular Material	Angular Material es una biblioteca de componentes de IU para desarrolladores de Angular. Los componentes de Material Angular ayudan a construir páginas web y aplicaciones web atractivas, consistentes y funcionales a la vez que se adhieren a los principios modernos de diseño web, como la portabilidad del navegador, la independencia del dispositivo y la degradación elegante [23].	Diseño de la Aplicación web

Nombre	Descripción	Utilizado en
 Bootstrap Bootstrap	<p>Bootstrap es un framework que se enfoca en la creación de sitios <i>responsive</i> para dispositivos móviles. Bootstrap tiene una colección de código reusable escrito en HTML, CSS y Javascript. Bootstrap se enfoca en el redimensionamiento de elementos como imágenes, cuadrículas, tablas, y muchos más elementos para que estos sean visualizados de una manera correcta en dispositivos móviles [24].</p>	<p>Diseño de la Aplicación web</p>
 git Git	<p>Es un sistema distribuido de control de versiones de código abierto y gratuito diseñado para manejar desde pequeños proyectos hasta muy grandes, con rapidez y eficiencia. Con el uso de ramas se pueden crear diferentes versiones del proyecto para que estas sean manipuladas y cambiadas sin problema [25].</p>	<p>Versionamiento de la Aplicación web</p>
 Github	<p>GitHub es un repositorio para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador [26].</p>	<p>Versionamiento de la Aplicación web</p>
 Node.js	<p>Node.js es un entorno de tiempo de ejecución multiplataforma de código abierto para desarrollar aplicaciones de red y del lado del servidor. Las aplicaciones Node.js están escritas en JavaScript y pueden ejecutarse dentro del tiempo de ejecución de Node.js en macOS, Microsoft Windows y Linux. Node.js también proporciona una rica biblioteca de varios módulos JavaScript que simplifica el desarrollo de aplicaciones web usando Node.js en gran medida. Node.js se puede usar para crear diferentes tipos de aplicaciones, como aplicaciones de línea de comandos, aplicaciones web, aplicaciones de chat en tiempo real, servidor API REST, etc. Sin embargo, se usa principalmente para crear programas de red como servidores web, similares a PHP, Java o ASP.NET [27].</p>	<p>Aplicación web</p>
 NPM	<p>Node Package Manager es un gestor de paquetes y un instalador. Además de eso es considerada una de las librerías más grandes de proyectos de código abiertos de node.js. Npm es una herramienta de línea de comandos para interactuar con los repositorios. Usando comandos npm se facilita la instalación de paquetes, administración de versiones, y administrar dependencias de los paquetes [28].</p>	<p>Aplicación web</p>

Nombre	Descripción	Utilizado en
 PrimeNG	PrimeNG es una colección de código abierto de componentes de interfaz de usuario para Angular [29].	Diseño de la Aplicación web
 Typescript	TypeScript es una extensión de JavaScript, diseñada para facilitar el desarrollo de Aplicaciones JavaScript a gran escala. TypeScript es tanto un lenguaje de código abierto como un conjunto de herramientas. Se puede ejecutar en Node.js o en cualquier navegador que admita ECMAScript 3 o versiones más recientes [30].	Aplicación web
 Visual Studio Code	Visual Studio Code es un editor de código optimizado con soporte para operaciones de desarrollo como depuración, ejecución de tareas y control de versiones. Su objetivo es proporcionar las herramientas que un desarrollador necesita para un ciclo rápido de creación de código-depuración y deja flujos de trabajo [31]. Visual Studio Code es de código abierto y es gratis para el uso privado y comercial. Incluye soporte para el desarrollo con Node.js usando JavaScript y TypeScript. Visual Studio Code también incluye herramientas para tecnologías web como JSX / React, HTML, CSS, SCSS, Less y JSON [32].	Desarrollo de la aplicación web

Tabla 3. Herramientas de Front-End. Elaborada por el autor.

2 METODOLOGÍA

En esta sección, se explicará la metodología utilizada durante el desarrollo de la aplicación. Se explicará la razón por la cual se escogió la metodología del prototipado en vez de otras metodologías existentes. A la vez profundizando que tipo de metodología es y como beneficia al proyecto.

2.1 Metodología del Prototipado

Para el desarrollo de este proyecto, se propone utilizar el prototipado como metodología de desarrollo. Esta metodología consiste en desarrollar versiones funcionales del programa obteniendo una retroalimentación rápida. Así, el objetivo principal del prototipado es la entrega rápida de valor para el usuario final, recibiendo rápidamente retroalimentación que permita comprender mejor los requisitos de software y así poder refinarlo en cada entrega hasta obtener un software completo y funcional [33]. Entre los beneficios de esta metodología están [34]:

- Facilita la obtención de requerimientos del software
- Agiliza el proceso de resolución de problemas del software
- Agiliza el proceso de diseño de interfaces de usuario
- Ayuda en la ejecución de pruebas y la detección de errores

Esta metodología es evolutiva ya que se hacen entregas en tiempo “cortos” enfocándose en la mejora continua del software. Esta es una opción preferida por clientes y desarrolladores ya que el usuario puede obtener una mejor idea de cómo va a ser el software, y el desarrollador logra crear el mismo de una manera más rápida [35].

Al buscar versiones funcionales del programa para obtener retroalimentación rápida [33], se implementan prototipos de baja, media y alta fidelidad. Las características que posee cada prototipo son [36]:

1. Baja Fidelidad: Se implementarán los aspectos generales del sistema sin entrar en detalles. Se realizan maquetas (mockups) para definir y emular la funcionalidad del sistema.
2. Media Fidelidad: Usando la retroalimentación recolectada sobre el prototipo de baja fidelidad, se creará un prototipo inicial. Este se realizará usando el framework de Angular para el desarrollo de la interacción entre el modelo, la vista y el controlador.
3. Alta Fidelidad: Usando la retroalimentación recolectada sobre el prototipo de media fidelidad, se presentará un sistema funcional a los usuarios finales para recibir de ellos una retroalimentación que permita decidir si el prototipo se desecha o se continúa trabajando sobre éste.

En la Figura 4 se puede ver el ciclo de vida de software basado en la metodología de prototipado la cual se va a implementar en este proyecto de titulación. La metodología empieza por el análisis de los requerimientos que son descripciones informales de las necesidades del usuario. Una vez obtenidos los requerimientos informales, estos se traducen para obtener la especificación y necesidades del sistema. Usando mockups se elaborará el prototipo de baja fidelidad o prototipo de interfaz de usuario.

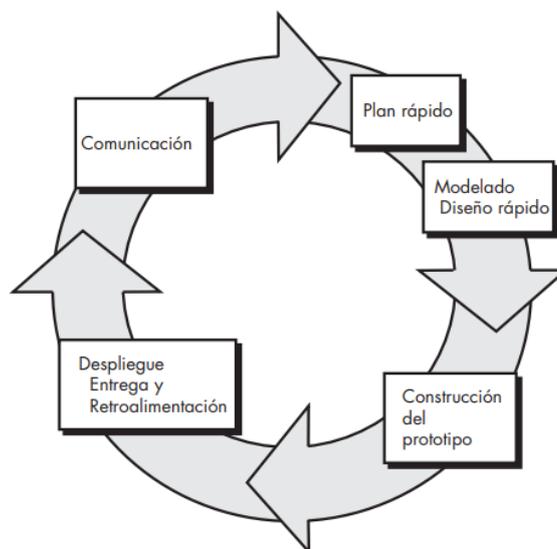


Figura 4. Ciclo de Vida del Software basado en Prototipado [35].

Los prototipos que se elaborarán seguirán el ciclo de refinamiento de prototipo que se puede visualizar en la Figura 5. A continuación, usando los requerimientos y objetivos, se escogerá qué tipo de prototipo se usará. Una vez especificado el tipo de prototipo que se va a usar, se generará un plan donde se especificarán las actividades y restricciones de tiempo para el desarrollo del prototipo. El prototipo será probado por el usuario y después evaluado por él cliente. Si este cumple con las expectativas del cliente se obtendrá un prototipo funcional. Si se encuentra cualquier error o falencia en cualquiera de estas etapas se podrá retroceder a etapas anteriores para hacer los cambios necesarios para obtener un prototipo aceptado. Se repetirán y mejorarán estos prototipos las veces que sean necesarias hasta obtener que los prototipos funcionales sean aceptados por el cliente.

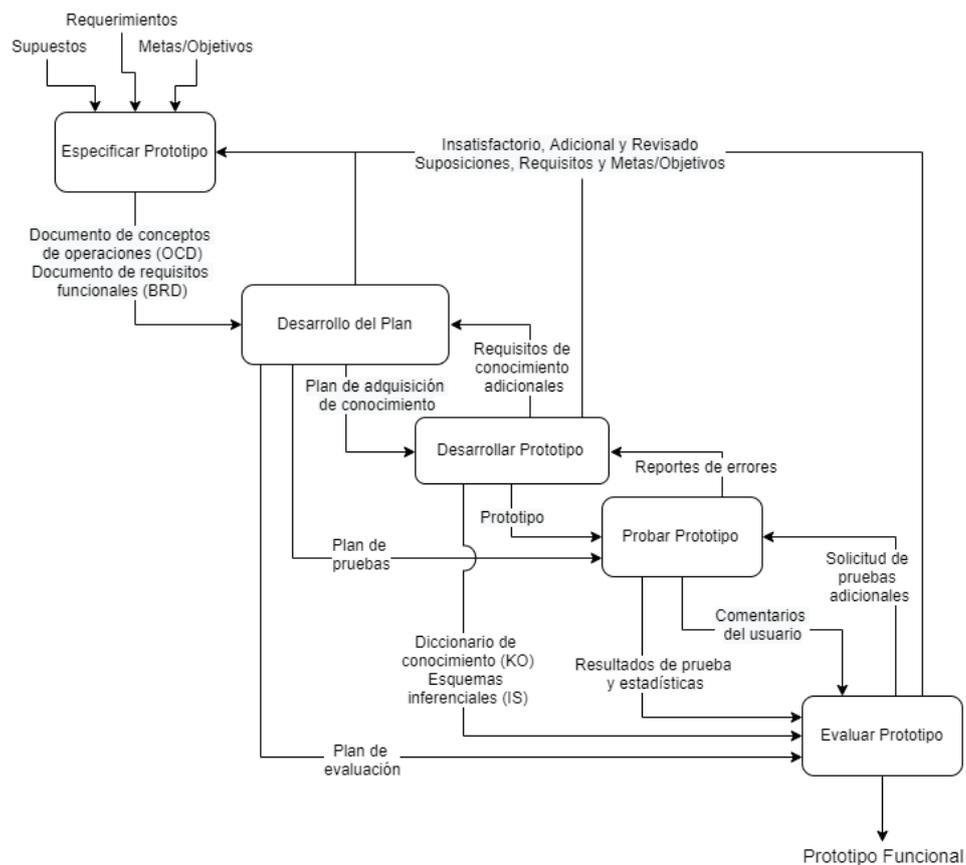


Figura 5. Ciclo de refinamiento de prototipo [38].

El prototipo de interfaz de usuario será evaluado y corregido hasta obtener la aceptación del cliente. En cada mejora y/o cambio se modificará y actualizará

los requerimientos definidos. Se continuará con el diseño de la arquitectura y componentes que formaran la parte lógica del sistema incluyendo la interacción entre el modelo, la vista, y el controlador. Usando este diseño se generará el prototipo de media fidelidad o prototipo de arquitectura y componentes. Esto se hará usando el framework de Angular donde se creará los modelos, las vistas y controladores con sus interacciones necesarias.

En la fase de implementación se formalizará el sistema y la documentación necesaria. Aquí se generará el sistema funcional donde se ejecutarán las pruebas de sistema para rechazar o aprobar el sistema. Si este fuera rechazado, dependiendo de las observaciones, se volverá a la fase que nos permitirá resolver los problemas encontrados. Si se encontraran nuevos requerimientos, estos serán añadidos a la definición de requerimientos. Si el sistema funcional fuera aprobado, se obtendría el producto final y se avanzará a la etapa final de operación y mantenimiento.

Se escogió esta metodología sobre las metodologías tradicionales ya que, si existe algún error o cambio grande después de cualquier revisión, esta no afecta al rendimiento del proyecto ya que el prototipado es una metodología ágil. No se escogió las metodologías ágiles populares como son XP o Scrum ya que el proyecto será desarrollado por una sola persona y estas metodologías están más orientadas a un equipo de trabajo con diferentes roles. Pero si se utilizará artefactos de XP, como las historias de usuario.

Las historias de usuario son un mecanismo usado en XP que ayuda a “definir una cosa que el cliente quiere que el sistema haga” [39]. Las historias ayudan a definir un punto de partida para el proyecto, ya que estas historias se pueden descomponer en tareas cuantificables que deben ser desarrolladas. Estas historias también deben ser estimadas en cuanto tiempo tomar en implementar cada tarea. Estas son las principales razones por la que se escogió usar las historias de usuario en este proyecto. Estas ayudarán a definir tareas concretas que se deberán completar para cada prototipo y estimar la dificultad de cada tarea para poder organizar cada historia por prioridad. Así generando un plan organizado para cada prototipo e iteración que se implementará.

2.2 Aplicación de la Metodología

2.2.1 Prototipo de Baja Fidelidad

2.2.1.1 Plan Rápido

Objetivos

- Recolectar requerimientos para así plasmar módulos e interfaces que serán diseñadas en el prototipo.
- Diseñar un prototipo de interfaces de usuario para simular la navegabilidad del producto final.

Historias de Usuario

Se recolectaron los siguientes requerimientos funcionales que son necesarios para la implementación del sistema. Cada requerimiento está definido como una historia de usuario que contiene su propio código, descripción, peso de complejidad, y prioridad. El peso de complejidad será definido por un número del uno al diez. Uno siendo sencillo o de complejidad baja y diez siendo complicado o de complejidad alta. La prioridad es la importancia que tiene la historia de usuario para el interesado y será definida como Baja, Media o Alta. A continuación, en la Tabla 4 se presenta el listado de historias de usuario que se han obtenido.

Código	Historia de usuario	Peso	Prioridad
US001	Se necesita una pantalla para el control de usuario antes de usar el sistema.	3	Alta
US002	Se necesita una pantalla para registrar usuarios	3	Alta
US003	Para iniciar sesión es necesario usar el nombre de usuario correspondiente a la cuenta.	3	Alta
US004	Se podrá visualizar los actores creados y sus datos como nombre, prioridad y comentario.	2	Alta
US005	Se podrá visualizar los temas creados y sus datos como nombre, prioridad y comentario.	2	Alta

US006	Se podrá crear una matriz con los actores y temas existentes.	3	Alta
US007	Se podrá visualizar los nombres de los temas y actores de la matriz.	3	Alta
US008	Se podrá marcar las celdas donde exista una interrelación entre tema y actor.	3	Alta
Total		22	

Tabla 4. Listado de Historias de Usuarios para el prototipo de baja fidelidad.
Elaborado por autor.

2.2.1.2 Desarrollo del Prototipo

Módulos

Usando las historias de usuarios se obtuvo los siguientes módulos del sistema:

- Módulo de Control de Usuario
- Módulo de Actores
- Módulo de Temas
- Módulo de Matriz

Construcción del Prototipo

Para la construcción del prototipo de baja fidelidad se usó la herramienta AdobeXD. El objetivo de este prototipo es mostrar visualmente como se verán las interfaces de usuario y simular la navegabilidad entre ellas. Este prototipo ayuda al usuario a entender cómo se vería y como se usaría la aplicación de una manera visual y gráfica. En la Figura 6 se muestra el plano general de las interfaces de usuarios y sus conexiones entre sí.

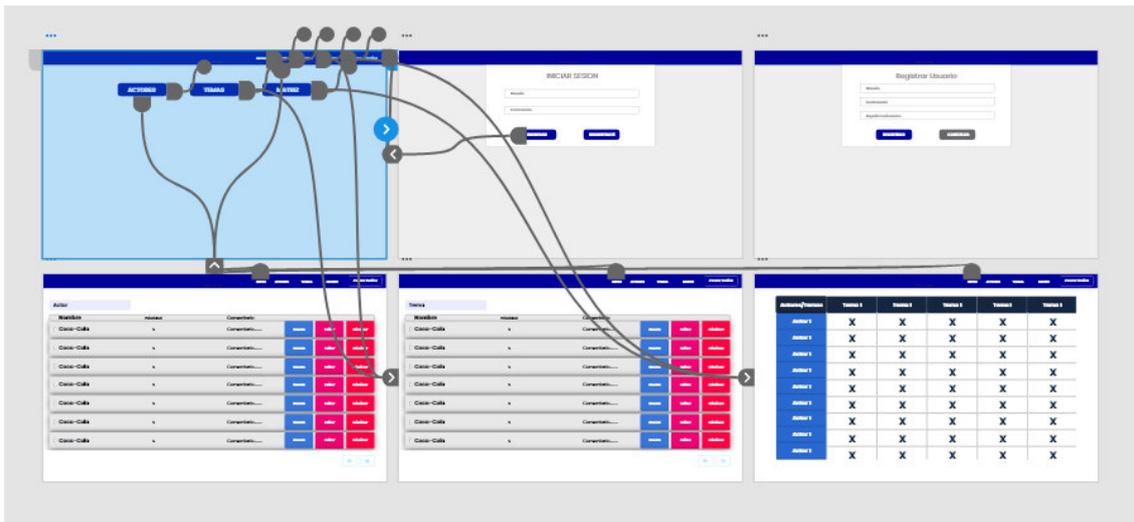


Figura 6. Vista general de las pantallas en AdobeXD. Elaborado por el autor.

Control de Usuario

Esta será la primera pantalla que será presentada al usuario al iniciar el sistema. En esta pantalla se tendrán dos opciones: la primera que permite ingresar las credenciales e iniciar la sesión, y la segunda que se permita iniciar el registro de un usuario nuevo. Se puede visualizar esta pantalla en la Figura 7.

Al pulsar el botón de *Regístrate*, el sistema llevará al usuario a la pantalla de registro de usuario. En esta pantalla se muestra un formulario donde se podrán llenar los datos referentes al nombre de usuario y contraseña. Datos adicionales podrán ser requeridos a futuro. Se puede visualizar esta pantalla de registro en la Figura 8.

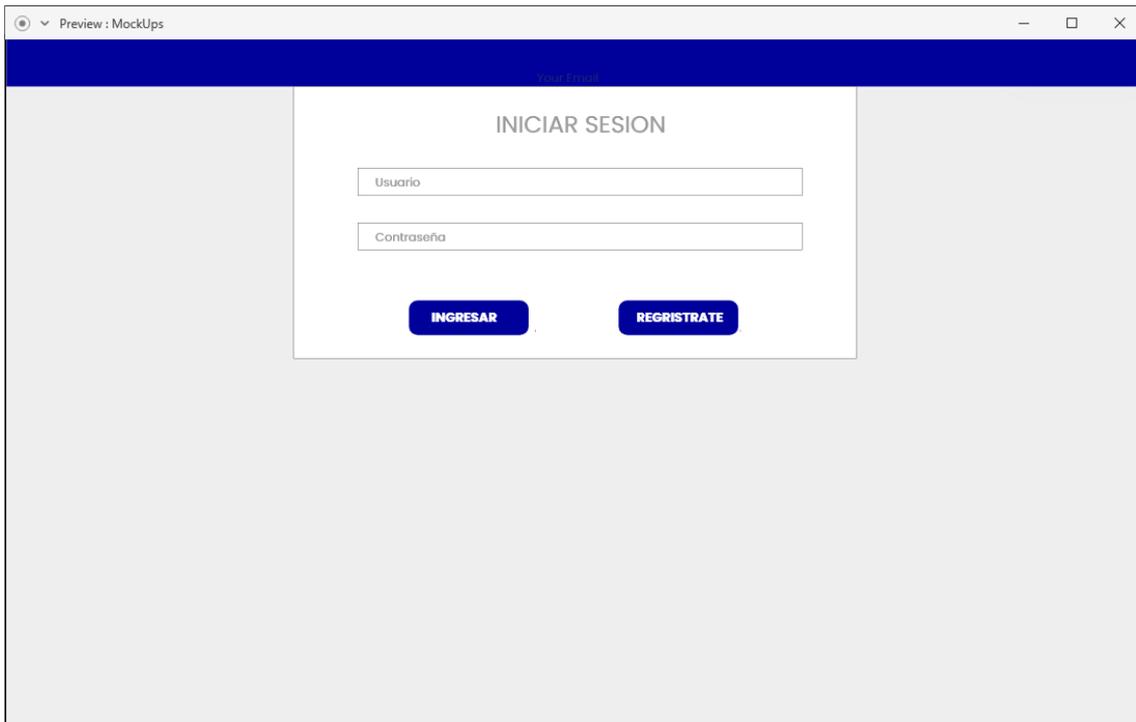


Figura 7. Pantalla de Inicio de Sesión - Prototipo Baja Fidelidad. Elaborado por el autor.

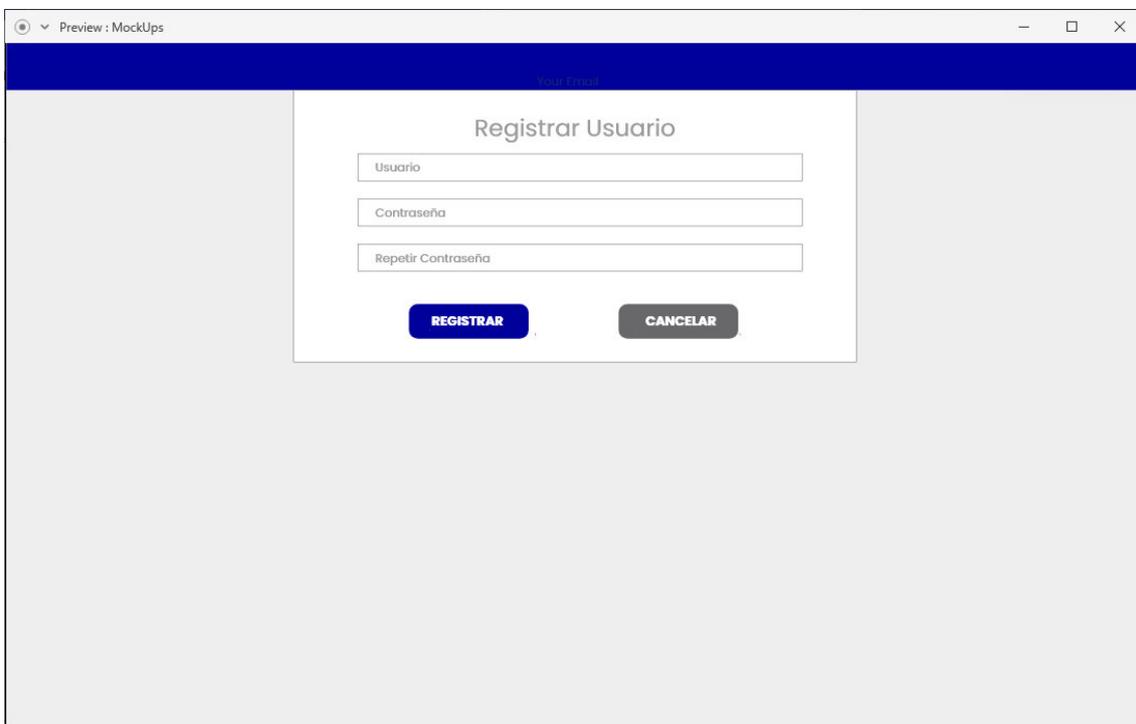


Figura 8. Pantalla de Registro de Usuario - Prototipo Baja Fidelidad. Elaborado por el autor

Inicio

Una vez completado el control de acceso se desplegará la pantalla de Inicio. En esta pantalla se mostrarán las opciones para navegar a las diferentes pantallas que se tienen en el sistema. Esto se puede hacer mediante el uso de la barra de navegación o los botones que se despliegan en la pantalla. Esta pantalla se puede visualizar en la Figura 9.

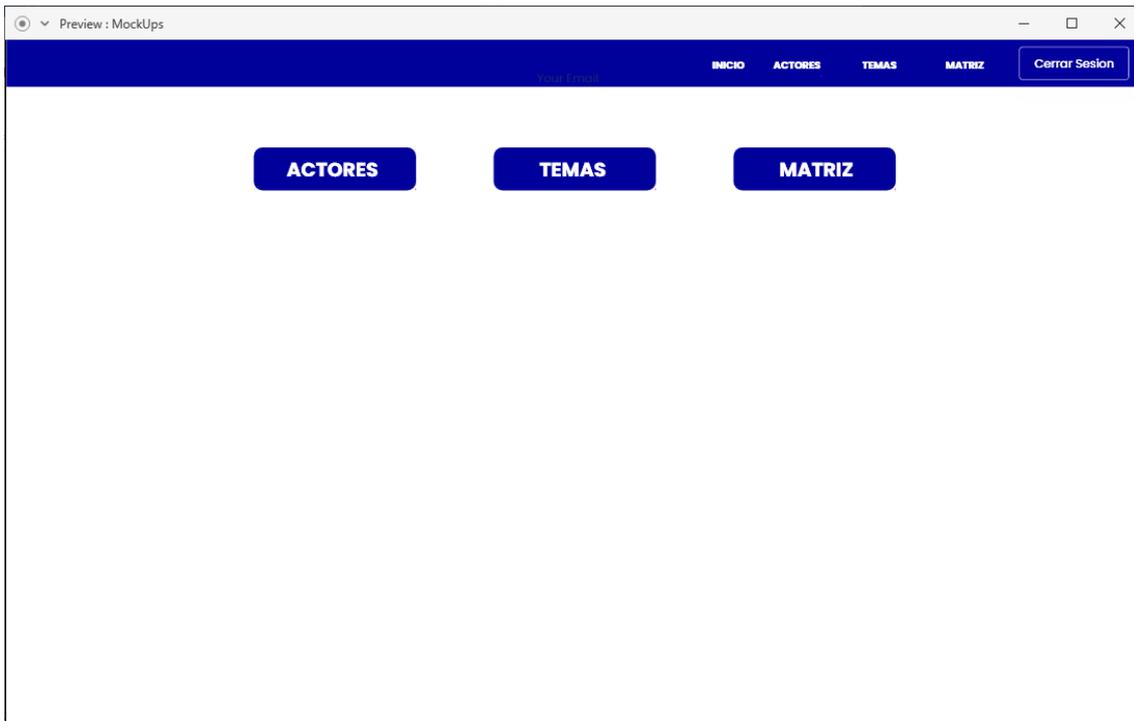


Figura 9. Pantalla de Inicio - Prototipo Baja Fidelidad. Elaborado por el autor

Actores

El objetivo de esta pantalla es mostrar los actores que han sido creados en el sistema en el contexto de SScan. Los actores deben incluir los siguientes datos: un nombre, una prioridad y un comentario. Cada actor creado se puede editar o eliminar. En esta pantalla también se debe poder crear nuevos actores. Esta pantalla se puede visualizar en la Figura 10.

Temas

El objetivo de esta pantalla es mostrar los temas que han sido creados en el sistema en el contexto de SScan. Los temas deben tener los siguientes datos: un nombre, una prioridad y un comentario. Cada tema creado se puede editar o

eliminar. En esta pantalla también se debe poder crear nuevos temas. Esta pantalla se puede visualizar en la Figura 11.

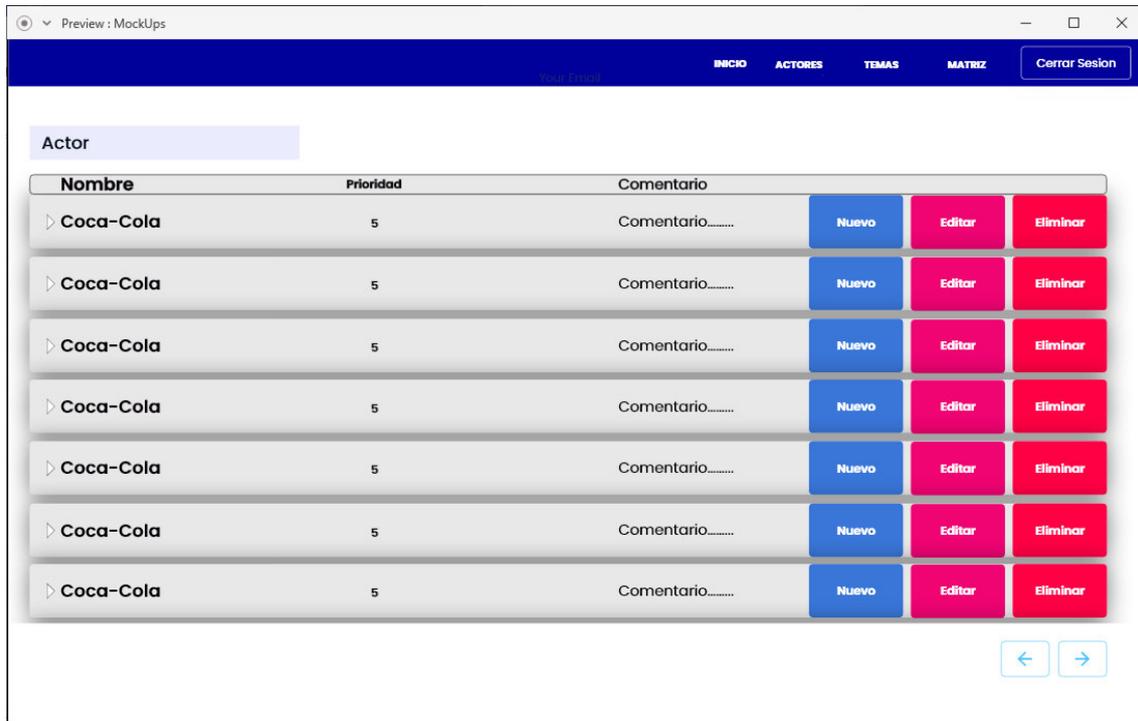


Figura 10. Pantalla de Actores - Prototipo Baja Fidelidad. Elaborado por el autor

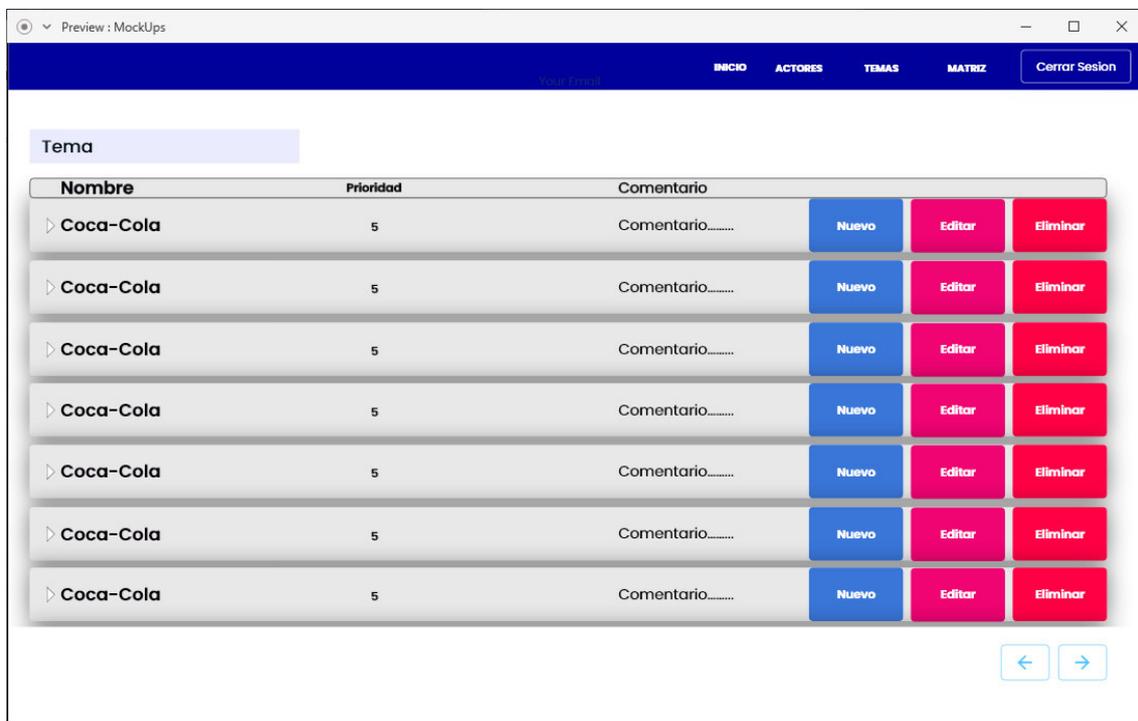
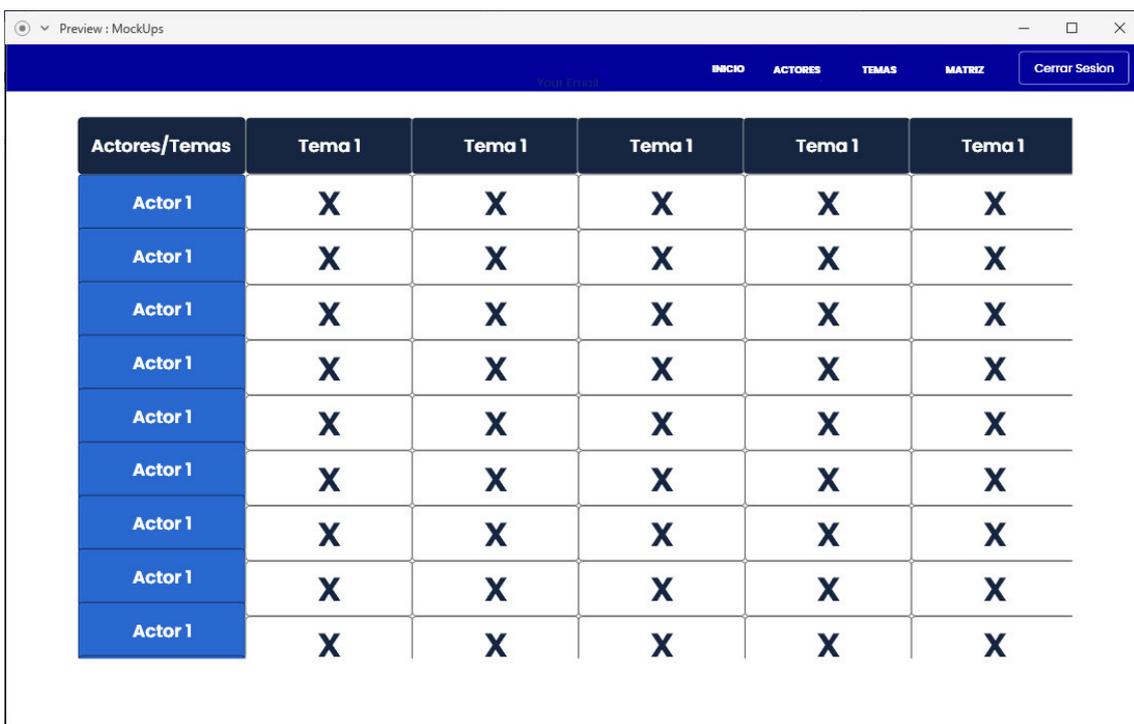


Figura 11. Pantalla de Temas - Prototipo Baja Fidelidad. Elaborado por el autor

Matriz

El objetivo de esta pantalla es mostrar una matriz creada a partir de los actores y temas que han sido creados en el sistema. La matriz debe mostrar los nombres individuales de cada actor y tema, creando así las cabeceras de las columnas y filas de la matriz. En cada celda creada se podrá crear una marca para mostrar que existe una interrelación entre ese tema y actor específico. Esta pantalla se puede visualizar en la Figura 12.



Actores/Temas	Tema 1				
Actor 1	X	X	X	X	X
Actor 1	X	X	X	X	X
Actor 1	X	X	X	X	X
Actor 1	X	X	X	X	X
Actor 1	X	X	X	X	X
Actor 1	X	X	X	X	X
Actor 1	X	X	X	X	X
Actor 1	X	X	X	X	X
Actor 1	X	X	X	X	X

Figura 12. Pantalla de Registro de Matriz - Prototipo Baja Fidelidad. Elaborado por el autor

2.2.1.3 Evaluación del prototipo

El desarrollo del prototipo de baja fidelidad se llevó a cabo sin mayores inconvenientes. Por esta razón no fue necesario actualizar el listado de historias de usuario.

2.2.2 Prototipo de Media Fidelidad

2.2.2.1 Plan Rápido

Objetivos

- Diseñar el modelo de datos que será utilizado para la base de datos.
- Diseñar la arquitectura que tendrá el sistema.
- Implementar el modelo de datos en Prisma.
- Construir el prototipo de media fidelidad usando Angular, basándose en el diseño presentado en el prototipo de baja fidelidad.
- Diseñar y probar los módulos implementados.

Historias de Usuario

En la Tabla 5, se presentan las nuevas historias de usuario que se necesitará implementar en este prototipo.

Código	Historia de usuario	Peso	Prioridad
US009	Desplegar el modelo de datos en el servidor Prisma.	5	Alta
US010	Levantar la arquitectura entre Angular, Prisma y la base de datos en PostgreSQL.	5	Alta
US011	Se necesita un control de usuario antes de usar el sistema.	3	Alta
US012	Para registrar un usuario es necesario un nombre de usuario.	3	Alta
US013	Se podrá cerrar la sesión una vez terminado el uso del sistema.	2	Media
US014	Se podrá visualizar los actores creados y sus datos como nombre, prioridad y comentario.	2	Alta
US015	Se podrá crear actores que incluyan un nombre, una prioridad, y un comentario que sea opcional.	3	Alta
US016	Se podrá editar cualquier dato de un actor existente.	4	Alta
US017	Se podrá eliminar cualquier dato de un actor existente.	3	Alta
US018	Se podrá visualizar los temas creados y sus datos como nombre, prioridad y comentario.	2	Alta
US019	Se podrá crear temas que incluyan un nombre, una prioridad, y un comentario que sea opcional.	3	Alta
US020	Se podrá editar cualquier dato de un tema existente.	4	Alta
US021	Se podrá eliminar cualquier dato de un tema existente.	2	Alta
US022	Construir una pantalla donde se desplegará una matriz con los actores y temas existentes	7	Alta
Total		48	

Tabla 5. Listado de Historias de Usuarios para el prototipo de media fidelidad. Elaborado por autor.

2.2.2.2 Desarrollo del Prototipo

Construcción del Prototipo

Para iniciar el desarrollo de este prototipo primero se generó un repositorio en la herramienta de GitHub. GitHub va a mantener el versionamiento del código y ayudará a mantener una copia de seguridad del código en la web también conocido como control de versiones en la metodología de XP [40]. En la Figura 13 se puede visualizar el proyecto creado.

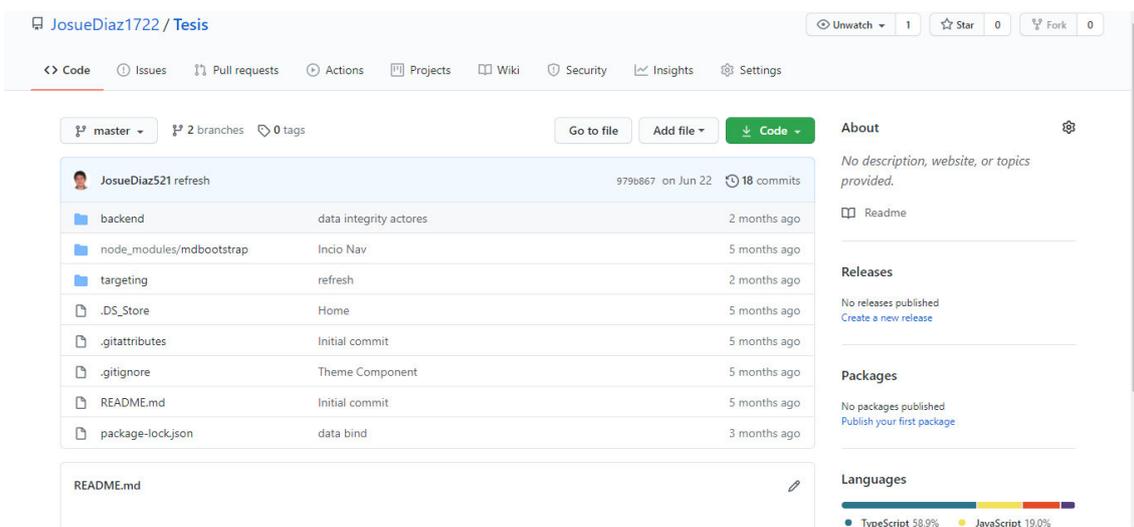


Figura 13. Repositorio creado en GitHub para el desarrollo del prototipo de media fidelidad. Elaborado por el autor.

Pantalla de inicio de Sesión

Esta pantalla es la primera pantalla que se muestra al iniciar el sistema. Aquí se tiene el formulario para llenar con las credenciales para entrar al sistema. También existe la opción de crear un nuevo usuario para los usuarios que no tengan las credenciales. Se puede visualizar esta pantalla en la Figura 14.

Inicio de session

Usuario *

Password *

Login

Crear Usuario

Figura 14. Pantalla de Inicio de Sesión. Elaborada por el autor.

Pantalla de Crear Usuario

En esta pantalla se tiene el formulario para llenar con las nuevas credenciales para entrar al sistema. También existe la opción de cancelar, que le redirige a la pantalla de inicio de sesión. Se puede visualizar esta pantalla en la Figura 15.

Inicio de session

Usuario *

Password *

Login

Crear Usuario

Figura 15. Pantalla de Crear Usuario. Elaborada por el autor.

Pantalla de Inicio

Es la pantalla principal del sistema donde se presenta al usuario con las opciones para navegar a las diferentes pantallas que ofrece el sistema, mediante los botones en la pantalla o la barra de navegación. Se puede visualizar esa pantalla en la Figura 16.

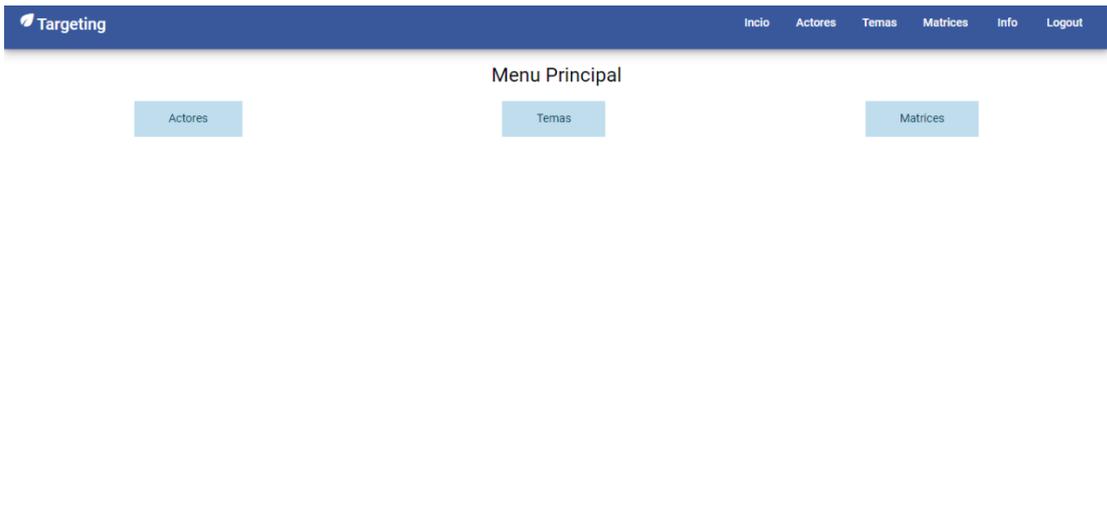


Figura 16. Pantalla de Inicio. Elaborada por el autor.

Pantalla de Actores

En esta pantalla se pueden visualizar los diferentes actores creados en el sistema. Esta pantalla tiene las opciones de editar, eliminar, crear un nuevo actor o crear un sub actor. Se puede visualizar esta pantalla en la Figura 17.

ActorID	Actor Name	Prioridad	Comentario
1	▼ Actor 1	4	Comentario
2	Sub Actor	4	Comentario Sub Actor
16	Actor 2	4	Test

localhost:4200/actores

Figura 17. Pantalla de Actores. Elaborada por el autor.

Pantalla de Temas

En esta pantalla se puede visualizar los diferentes temas creados en el sistema. Esta pantalla tiene las opciones de editar, eliminar, crear un nuevo tema o crear un subtema. Se puede visualizar esta pantalla en la Figura 18.

Targeting				Inicio	Actores	Temas	Colaborativo	Info	List	List
Nuevo Tema										
TemaID	Tema Name	Prioridad	Comentario							
3	▼ Tema	5		[User] [Edit] [Delete]						
5	▼ sub Tema 1	3	prueba	[User] [Edit] [Delete]						
9	Test1	5		[User] [Edit] [Delete]						
4	▼ Tema 2	2		[User] [Edit] [Delete]						
8	Sub Tema 2	5	3	[User] [Edit] [Delete]						
11	actor2	2		[User] [Edit] [Delete]						
< < 1 > >										1 of 1 pages (6 items)

Figura 18. Pantalla de Temas. Elaborada por el autor.

Pantalla de Matriz

En esta pantalla se creará la matriz con los temas y actores existentes. En las cabeceras horizontales se coloca los nombres de los temas existentes y en las cabeceras verticales los nombres de los actores existentes.

En cada celda creada por la matriz se crean dos grupos de botones. El primer grupo de botones, de tipo semáforo, permite cambiar la prioridad de la interrelación entre actores y temas. Verde siendo prioridad baja, amarillo prioridad media y rojo prioridad alta. El segundo grupo de botones representa el impacto en el tiempo que va de bajo, medio, a alto. Se puede visualizar esta pantalla en la Figura 19.

Targeting				Inicio	Actores	Temas	Colaborativo	Info	List	List
Table										
Actores/Temas	Tema sub Tema 1 Test1	Tema 2 Sub Tema 2	actor2							
Actor 1 Sub Actor	ckb4ci13x00az0725anyt1qcy [Green] [Yellow] [Red] Prioridad: Media [Low] [Med] [High] Tiempo:	[Green] [Yellow] [Red] Prioridad: Alta [Low] [Med] [High] Tiempo:	[Green] [Yellow] [Red] Prioridad: [Low] [Med] [High] Tiempo: Bajo							
Actor 2	[Green] [Yellow] [Red] Prioridad: [Low] [Med] [High] Tiempo: Alto	[Green] [Yellow] [Red] Prioridad: Baja [Low] [Med] [High] Tiempo: Medio	[Green] [Yellow] [Red] Prioridad: [Low] [Med] [High] Tiempo:							
				[Low] [Med] [High] Tiempo:						

Figura 19. Pantalla de Matriz. Elaborada por el autor.

2.2.2.3 Evaluación del Prototipo

El desarrollo del prototipo de mediana fidelidad se desarrolló sin mayor inconveniente. En este caso, sí hubo observaciones, las cuales fueron transformadas a historias de usuario (Tabla 6).

Estas nuevas historias se van a tomar en cuenta en el desarrollo del prototipo de alta fidelidad. Se toma esta decisión ya que los cambios requeridos son más de forma y no de desarrollo.

Código	Historia de usuario	Peso	Prioridad
US023	Agregar un símbolo para diferenciar Temas padres de Temas hijos.	2	Alta
US024	Agregar un símbolo para diferenciar Actores padres de Actores hijos.	2	Alta
US025	En las tablas de Actores y Temas colocar una calificación de estrellas para expresar la prioridad.	4	Alta
US026	Crear una pantalla de matrices para poder tener y operar varias matrices al mismo tiempo.	7	Alta
Total		15	

Tabla 6. Listado de Historias de Usuarios para el prototipo de alta fidelidad después de observaciones. Elaborado por autor.

2.2.3 Prototipo de Alta Fidelidad

2.2.3.1 Plan Rápido

Objetivos

- Implementar los cambios adquiridos en la evaluación del prototipo de media fidelidad.
- Construir el prototipo de alta fidelidad usando Angular, basándose en el diseño presentado en el prototipo de media fidelidad.
- Diseñar y probar los módulos implementados.

Historias de Usuario

En la Tabla 7 se presentan las nuevas historias de usuarios que se necesitará implementar en este prototipo.

Código	Historia de usuario	Peso	Prioridad
US027	Agregar un botón para mostrar que no se ha escogido prioridad o tiempo en la matriz.	2	Alta
US028	Agregar un filtro por prioridad de actor y tema en la matriz.	8	Alta
US029	Agregar una lista de matrices disponibles y una lista de las matrices que el usuario ha creado	6	Alta
US030	Agregar estilos css para la matriz creada.	2	Alta
Total		18	

Tabla 7. Listado de Historias de Usuarios para el prototipo de alta fidelidad.
Elaborado por autor.

2.2.3.2 Desarrollo del Prototipo

Módulos

Usando las historias de usuarios se obtuvo que se necesitaría crear el siguiente módulo del sistema:

- Módulo de Matrices

Construcción del Prototipo

Pantalla de Actores

En esta pantalla se pueden visualizar los diferentes actores creados en el sistema. Se puede ver que se ha hecho el cambio descrito en la historia de usuario de código US023, donde se especifica la necesidad de un ícono para diferenciar padres e hijos. Se puede notar, además, el cambio referente a la historia de usuario de código US025, donde se especifica la necesidad de una calificación tipo estrellas para la prioridad de los actores. Se puede visualizar esta pantalla en la Figura 20.

Targeting			Inicio	Actores	Temas	Matrices	Info	Logout
								Nuevo Actor
Actor	Prioridad	Comentario						
▼ Actor 2	☆☆☆		+u	✎	🗑			
↳ Sub Actor 2,1	☆☆☆		+u	✎	🗑			
▼ Actor 3	☆☆☆		+u	✎	🗑			
↳ Sub Actor 3,1	☆☆☆		+u	✎	🗑			
↳ Sub Actor 3,1,1	☆☆☆		+u	✎	🗑			
↳ Sub Actor 3,1,1,1	☆☆☆		✎	🗑				
↳ Sub Actor 3,1,1,2	☆☆☆		✎	🗑				
Actor 4	☆☆☆		+u	✎	🗑			
▼ Actor 5	☆☆☆		+u	✎	🗑			
localhost:4200/actores	☆☆☆		+u	✎	🗑			

Figura 20. Pantalla de Actores. Elaborada por el autor.

Pantalla de Temas

En esta pantalla se puede visualizar los diferentes temas creados en el sistema. Se puede ver que se ha hecho el cambio descrito en la historia de usuario de código US024, donde se especifica la necesidad de un ícono para diferenciar padres e hijos. También, se puede notar el cambio referente a la historia de usuario de código US025, donde se especifica la necesidad de una calificación tipo estrellas para la prioridad de los temas. Esta pantalla se muestra en la Figura 21.

Targeting			Inicio	Actores	Temas	Matrices	Info	Logout
								Nuevo Tema
Tema	Prioridad	Comentario						
▼ Tema 1	☆☆☆		+u	✎	🗑			
↳ Sub Tema 1,1	☆☆☆		+u	✎	🗑			
↳ Sub Tema 1,2	☆☆☆		+u	✎	🗑			
▼ Tema 2	☆☆☆		+u	✎	🗑			
↳ Sub Tema 2,1	☆☆☆		+u	✎	🗑			
↳ Sub Tema 2,1,1	☆☆☆		+u	✎	🗑			
Tema 3	☆☆☆		+u	✎	🗑			
▼ Tema 4	☆☆☆		+u	✎	🗑			
↳ Sub Tema 4,1	☆☆☆		+u	✎	🗑			

Figura 21. Pantalla de Temas. Elaborada por el autor.

Pantalla de Matrices

En esta pantalla se permite crear la matriz con los temas y actores existentes; y visualizar todas las matrices creadas en el sistema, que se despliegan en la lista del lado izquierdo. Esta lista mostrará únicamente las matrices que ha creado el usuario que está conectado. Solo las matrices creadas por el usuario podrán ser eliminadas por el usuario que las creó. Se puede visualizar esta pantalla en la Figura 22.

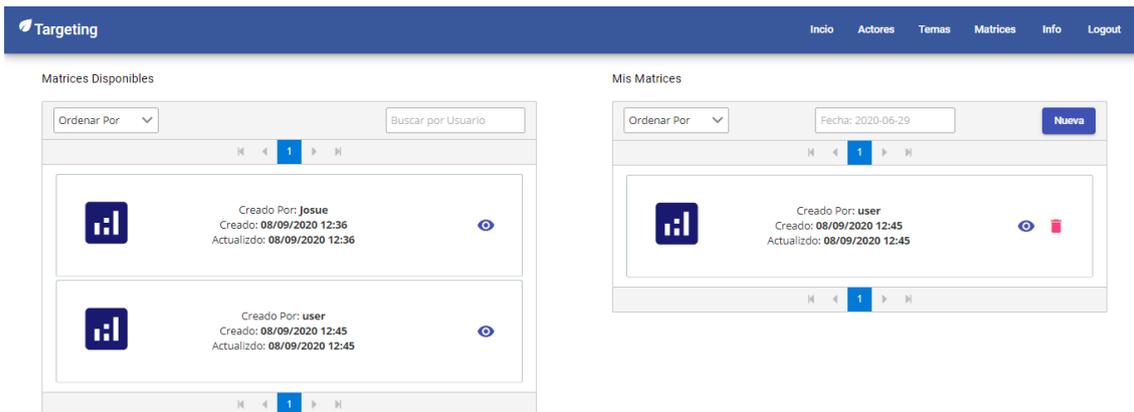


Figura 22. Pantalla de Matrices. Elaborada por el autor.

Pantalla de Matriz

En esta pantalla se observa la matriz que fue seleccionada en la pantalla de matrices. En las cabeceras horizontales se colocan los nombres de los temas existentes y en las cabeceras verticales los nombres de los actores existentes.

En cada celda creada por la matriz contendrá dos grupos de botones. El primer grupo de botones de tipo semáforo son para cambiar la prioridad de esa interrelación: verde siendo prioridad baja, amarillo prioridad media y rojo prioridad alta. El segundo grupo de botones representa el impacto en el tiempo que va de bajo, medio, a alto. A los dos grupos se les añadió un botón de no aplica para mostrar que en esta celda no se ha seleccionado prioridad o impacto en el tiempo.

En la parte superior de la matriz se agregó el botón de refrescar para poder refrescar la matriz para el uso de tiempo real con colaboradores. Adicionalmente se añadieron dos filtros uno para la prioridad de actor y la prioridad de tema. Cuando una de las celdas cumpla con la condición puesta en los dos filtros, ésta cambiará de color. Se puede visualizar esta pantalla en la Figura 23.

Actores/Temas		Tema 1		Tema 2		Tema 3		Tema 4				
		Sub Tema 1,1	Sub Tema 1,2	Sub Tema 2,1,1	Sub Tema 2,1,1		Sub Tema 4,1					
Actor 2	Sub Actor 2,1	N/A	●	●	●	●	N/A	●	●	●	●	
		N/A	●	●	●	●	N/A	●	●	●	●	
Actor 3	Sub Actor 3,1	Sub Actor 3,1,1	N/A	●	●	●	●	N/A	●	●	●	●
			N/A	●	●	●	●	N/A	●	●	●	●
	Sub Actor 3,1,2	N/A	●	●	●	●	N/A	●	●	●	●	
		N/A	●	●	●	●	N/A	●	●	●	●	
Actor 4		N/A	●	●	●	●	N/A	●	●	●	●	

Figura 23. Pantalla de Matriz. Elaborada por el autor.

2.2.3.3 Evaluación del Prototipo

El desarrollo del prototipo de alta fidelidad se llevó a cabo sin inconvenientes. Por esta razón no fue necesario actualizar las historias de usuario

2.3 Metodología de Evaluación

Para analizar el uso y comportamiento, grado de aceptación y facilidad de uso del Sistema, se utilizó el Modelo de Aceptación de la Tecnología (Technology Acceptance Model, TAM). TAM sugiere que la utilidad y la facilidad de uso predicen la usabilidad del sistema a través de variables mediadoras de actitud e intención [41]. El uso de la metodología de prototipado que mide la aceptación y usabilidad como criterio de evaluación coincide con el objetivo de la metodología TAM, por esta razón se decidió usar TAM como metodología de evaluación.

TAM se considera la teoría más influyente y comúnmente empleada para describir la aceptación individual de los sistemas de información [42]. Este modelo ayuda a explicar el uso de cierta tecnología, definiendo la aceptación de un sistema de información mediante cuatro variables [42] (Figura 24):

- La usabilidad percibida
- La facilidad de uso percibida
- La actitud hacia el uso.
- La intención de uso.

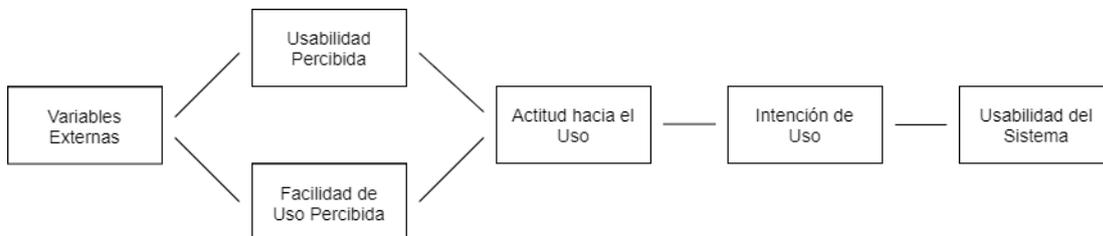


Figura 24. Modelo TAM [42].

La usabilidad percibida es definida como el grado en que una persona cree que un sistema de información particular mejorará su desempeño laboral. La facilidad de uso percibida es definida como el grado en que una persona cree que usar un sistema en particular es libre de esfuerzo. La actitud hacia el uso es definida como la evaluación del usuario sobre la conveniencia de emplear una aplicación de un sistema de información en particular. Por último, la intención de uso es definida como una medida de la probabilidad de que una persona ocupe la aplicación [43] [44].

TAM pretende que si una tecnología o innovación mejora el rendimiento de una persona y no aumenta en gran medida el esfuerzo requerido para realizar una función, el sistema se considera útil y fácil de usar y la persona tendrá más probabilidades de adoptar la tecnología, el servicio o el comportamiento [45]. La metodología de TAM puede aportar al sistema ayudando en la identificación de errores o molestias que pueden sufrir los usuarios ya que este sistema está enfocado a la colaboración de usuarios con múltiples entornos y habilidades. TAM ayudará a identificar una medida de usabilidad del sistema para así poder decidir si el sistema está funcionando correctamente o si se necesita alguna modificación. [8]

3 RESULTADOS

En esta sección, se explicarán los resultados obtenidos después de aplicar la metodología para el desarrollo de la aplicación. Se explicará la arquitectura y funcionamiento de la versión funcional del sistema. Igualmente, se mostrará los resultados de las pruebas de aceptación, siguiendo el modelo TAM, que fueron obtenidas.

3.1 Descripción del Sistema

3.1.1 Descripción de Arquitectura

Base de Datos

Para la creación de la base de datos se generó un modelo lógico que se puede visualizar en la Figura 25. Este modelo fue implementado en GraphQL que usa un tipo de datos llamados 'types', que equivale a las tablas en el modelo lógico.

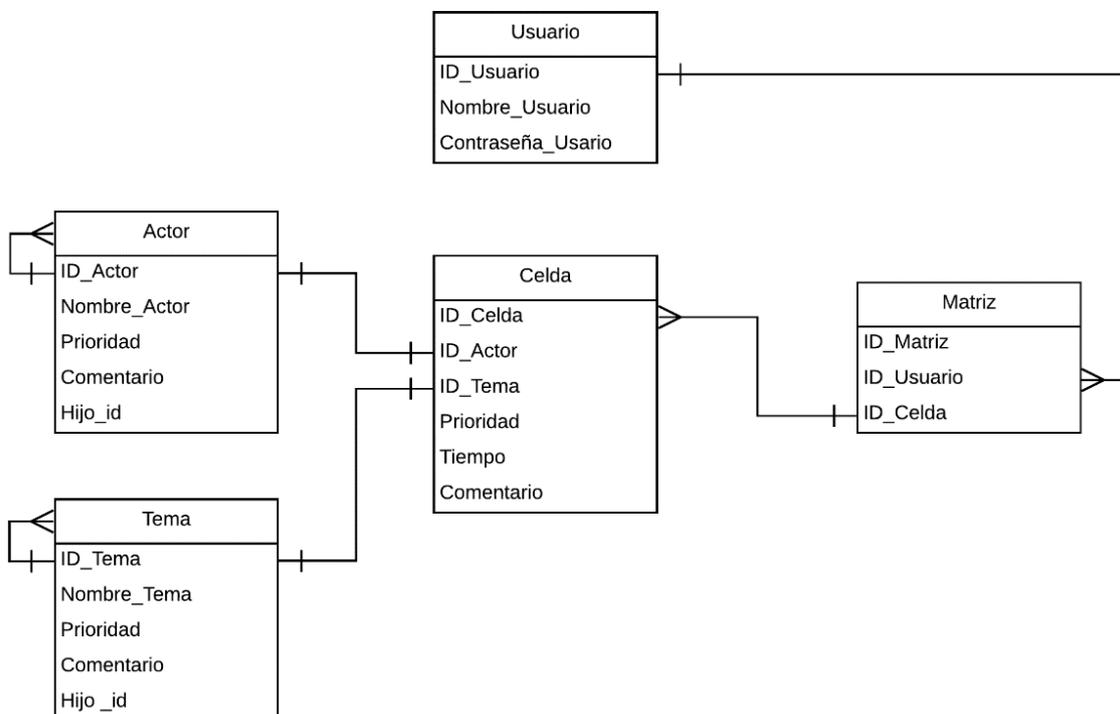


Figura 25. Modelo lógico de la base de datos. Elaborado por el autor.

Usando Prisma, que crea un servidor GraphQL y una conexión con PostgreSQL, se puede hacer un despliegue del modelo de datos para generar la base de datos y modelo de datos en GraphQL. Este despliegue se puede observar en la Figura 26.

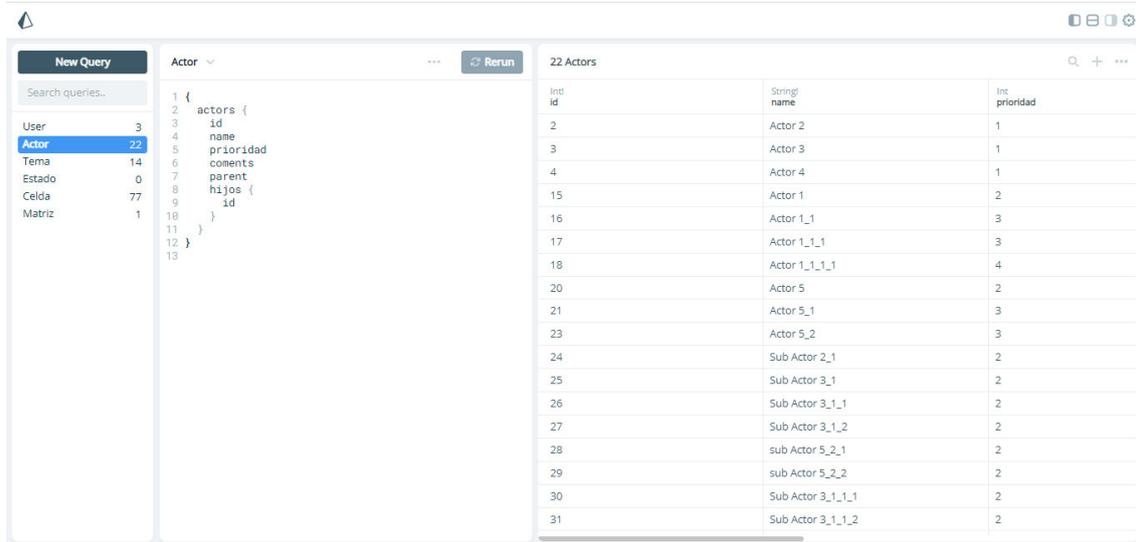


Figura 26. Modelo de datos desplegado con Prisma. Elaborado por el autor.

Arquitectura del Sistema

El sistema tiene una arquitectura de tipo Cliente-Servidor. Esta arquitectura está diagramada en la Figura 27. Se escogió esta arquitectura para facilitar el uso colaborativo del sistema ya que existirán varios usuarios que harán diferentes peticiones a un solo servidor.

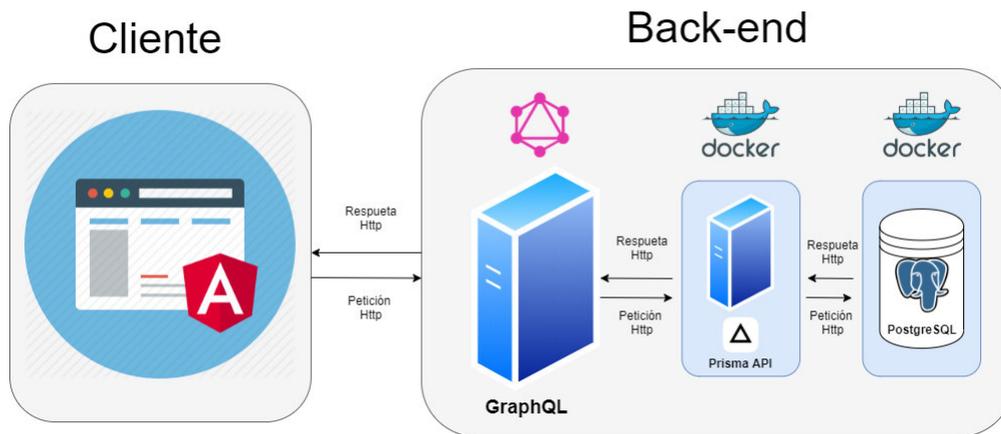


Figura 27. Diagrama de la arquitectura de la aplicación web. Elaborado por el autor.

La arquitectura es bidireccional. El cliente a través de peticiones HTTP, usando el internet, se comunica con el servidor GraphQL y este traslada la petición al servidor prisma cual transforma la consulta en idioma de GraphQL a una consulta de PostgreSQL. Una vez obtenida la información de la base de datos, ésta será trasladada al servidor de prisma que lo traslada al servidor de GraphQL el cual transforma la información a data con formato de objeto para así ser trasladado al cliente y desplegado o manipulado.

Subscriptores y Tiempo Real

Con el uso de GraphQL se puede implementar un método propio llamado subscriptores. GraphQL tiene tres tipos de operación, la primera corresponde a los queries que son usados para consultas de datos, el segundo se refiere a las mutaciones que son usadas para actualizar datos, y por último tenemos los subscriptores. Los subscriptores tienen la misma función que las consultas en el sentido de que sirven para recuperar datos almacenados, pero a diferencia de las consultas que crean una conexión con el servidor la cual se destruye al momento que se acaba la transacción; los suscriptores mantienen una conexión activa con el servidor que les informa sobre la acción que se toma [46]. Esto es posible con el uso de WebSockets.

Como se necesita una conexión activa que dé a conocer al servidor a todos los clientes conectados mediante los suscriptores, es necesario el uso del protocolo de WebSocket que nos provee una comunicación bidireccional que opera a través de un socket en la web y puede ayudar a crear una aplicación escalable y con una funcionalidad de tiempo real. El protocolo de WebSocket consiste en dos tipos de comunicaciones. El primero es el “handshake” o saludo que consiste en un mensaje del cliente y un mensaje de respuesta del servidor. El segundo permite, una vez que se creó el canal de comunicación, la transferencia de datos hasta que el canal es cerrado por el servidor o el cliente. Esta comunicación se puede visualizar en la Figura 28. Los WebSockets simplifican gran parte de la complejidad en torno de una comunicación web bidireccional y la gestión de conexiones. Esta tecnología permite además la sincronización en tiempo real entre múltiples usuarios a través de Internet y mediante un navegador web [47].

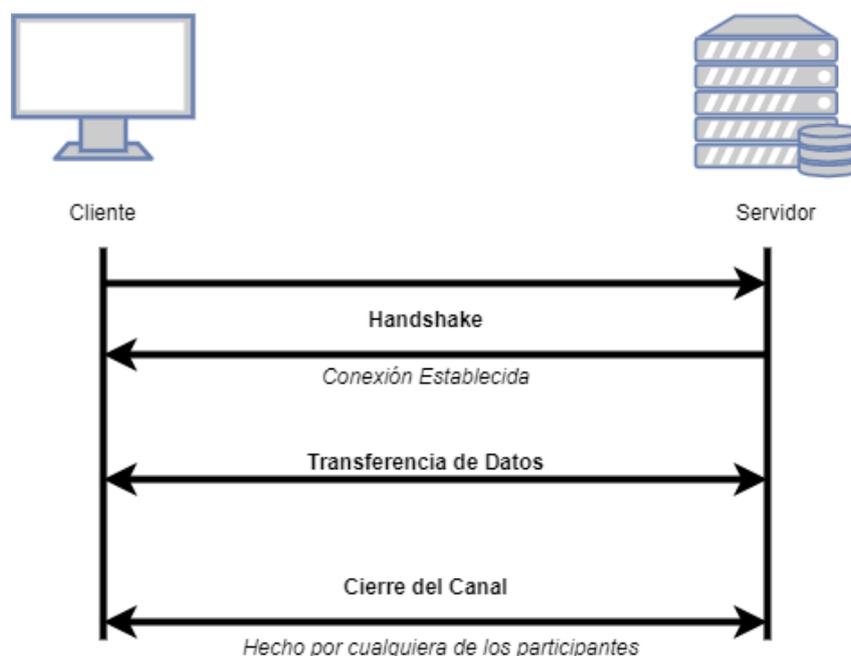


Figura 28. Protocolo de WebSocket. Elaborado por el autor.

Con la ayuda de los suscriptores, varios clientes pueden recibir los cambios que se han hecho ya que el servidor GraphQL notifica a los clientes de estos. Para poder implementar los suscriptores se necesita crear un link y establecer el puerto por donde se va a realizar esta conexión. GraphQL, por defecto, usa

el puerto 4466 para su comunicación, por esta razón la conexión de WebSocket tiene que ser hecha en este mismo puerto. Esto se puede visualizar en la Figura 29 donde, además de la dirección y el puerto, se especificó un tiempo alto de espera del puerto, para asegurar la conexión, aunque algún cliente tenga una conexión lenta de internet.

```
const ws = new WebSocketLink({
  uri: `ws://localhost:4466`,
  options: {
    reconnect: true,
    timeout: 30000
  }
});
```

Figura 29. Link de WebSocket. Elaborado por el autor.

Para definir un suscriptor es necesario establecer la entidad que se va a estar monitoreando, la función que se está monitoreando, el id de la entidad, y la información que se va a transmitir. Un ejemplo de un suscriptor se puede ver en la Figura 30.

```
subscription cambioMatriz ($id:ID!) {
  celda(
    where: {
      mutation_in: [UPDATED]
      node: {
        matriz: {
          id: $id
        }
      }
    }
  ) {
    mutation
    node {
      id
      prioridad
      tiempo
      TemaParent{id}
      ActorParent{id}
      matriz{id}
    }
  }
}
```

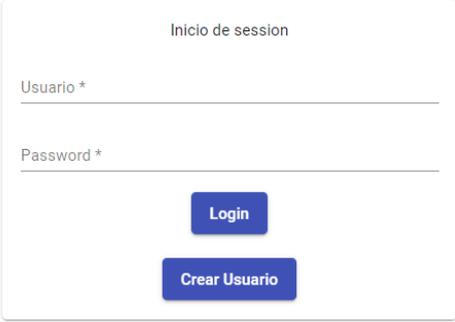
Figura 30. Ejemplo de un subscriptor. Elaborador por el autor.

En este ejemplo podemos ver que como parámetro se está mandando el id de la matriz. En la entidad celda se van a monitorear todas las actualizaciones que se haga, pero solo a las celdas que pertenezcan a la matriz que coincida con el id que se manda como parámetro. Para finalizar, los datos que se van a enviar a los clientes que estén conectados a este subscriptor van a ser el id, prioridad, tiempo, el id del tema, el id del actor, y el id de la matriz. Una vez hecho esto en nuestra pantalla de matriz, cuando en una celda sea actualizada la prioridad o el tiempo, a todos los clientes que estén viendo esa información se les actualizará los cambios hechos de una manera transparente y podrán visualizar estos inmediatamente, logrando así la interacción en tiempo real que se necesitaba para este sistema.

3.1.2 Descripción de la Aplicación Web

Pantalla de Login/Registro

El propósito de este aplicativo web es brindar una experiencia sencilla y eficaz, incluyendo un ámbito colaborativo para los usuarios. Al ingresar a la URL de la aplicación se encontrará un formulario de login que se puede visualizar en la Figura 31.

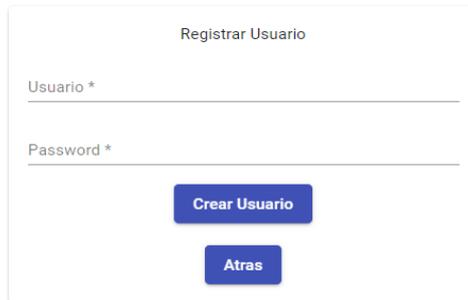


The image shows a web form for user login and registration. At the top, it says "Inicio de session". Below that, there are two input fields: "Usuario *" and "Password *". Underneath the "Password *" field, there are two buttons: "Login" and "Crear Usuario".

Figura 31. Pantalla inicial de la aplicación web. Elaborado por el autor.

Una vez ingresado al sitio, con el botón de “Crear Usuario” se accede a la sección de registros de usuario (Figura 32). Al enviar el formulario lleno, la

aplicación web dirigirá al formulario de Login para iniciar sesión con las credenciales con las que se registró.



Registrar Usuario

Usuario *

Password *

Crear Usuario

Atras

Figura 32. Página de Registro de Usuario de la aplicación web. Elaborado por el autor.

Pantalla de Inicio

Iniciada la sesión, se nos abrirá la página principal del sistema que se puede visualizar en la Figura 33. En esta página se puede navegar a los diferentes componentes del sistema que incluyen actores, temas, y las matrices colaborativas.

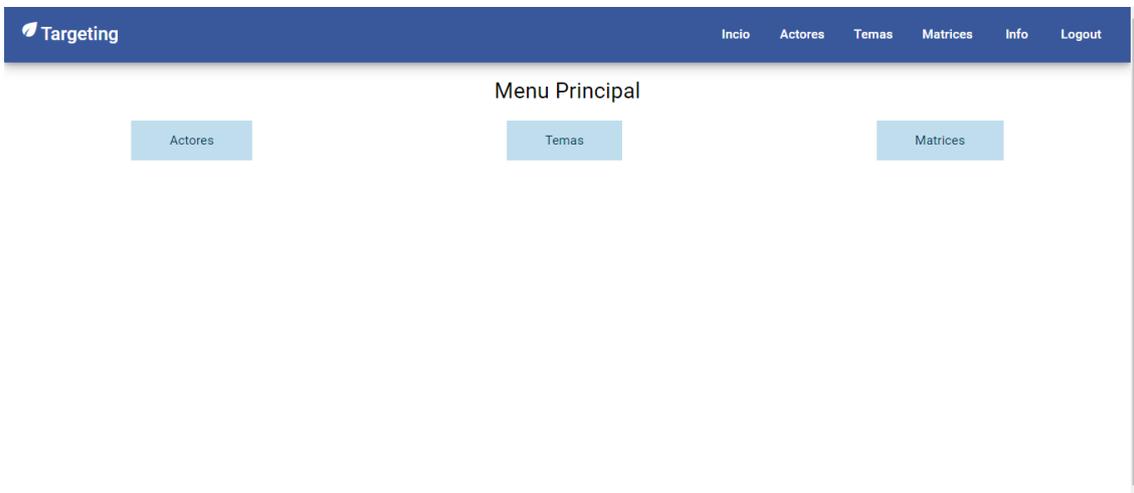


Figura 33. Página Principal de la aplicación web. Elaborado por el autor.

Pantalla de Actores

Como paso inicial, se necesitará definir actores principales que puedan o no tener sub-actores debajo de ellos. En la Figura 34 se puede visualizar la página de actores. Para poder definir los actores primero pulsamos el botón de actores sea en la pantalla inicial o la barra de navegación.



Figura 34. Página de Actores de la aplicación web. Elaborado por el autor.

Primero se tendrá que crear los actores padres que son los actores con números en la lista anteriormente creada. Para crear un padre se necesita pulsar el botón que se encuentra en la parte superior izquierda de la pantalla llamado “Nuevo Actor”. Una vez esto se desplegará el formulario de creación que se visualiza en la Figura 35.

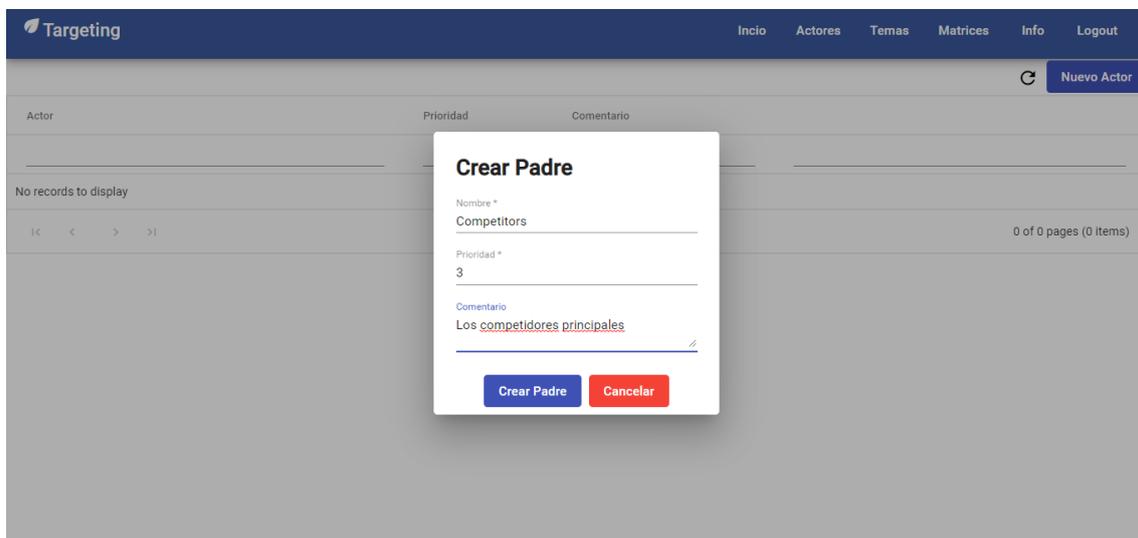


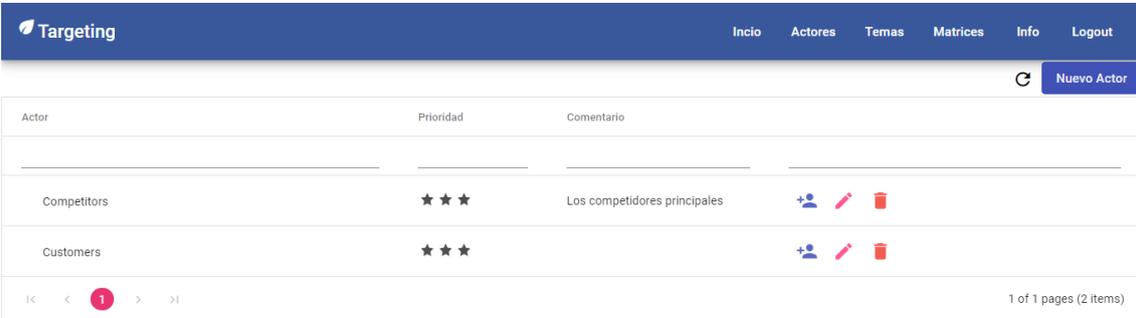
Figura 35. Formulario de Creación de Actor tipo Padre de la aplicación web. Elaborado por el autor.

Aquí se podrán ingresar los siguientes datos:

- Nombre: Tipo Texto
- Prioridad: Numero del 1 al 3
- Comentario: Tipo Texto o vacío

Una vez ingresado los datos se procederá a pulsar el botón “Crear Padre”.

Una vez creado el actor se mostrará en la lista de actores (Figura 36). Para crear un sub-actor dentro del actor “Customers”, por ejemplo, se pulsará el botón  en la fila correspondiente al actor escogido.



Actor	Prioridad	Comentario	
Competitors	★ ★ ★	Los competidores principales	  
Customers	★ ★ ★		  

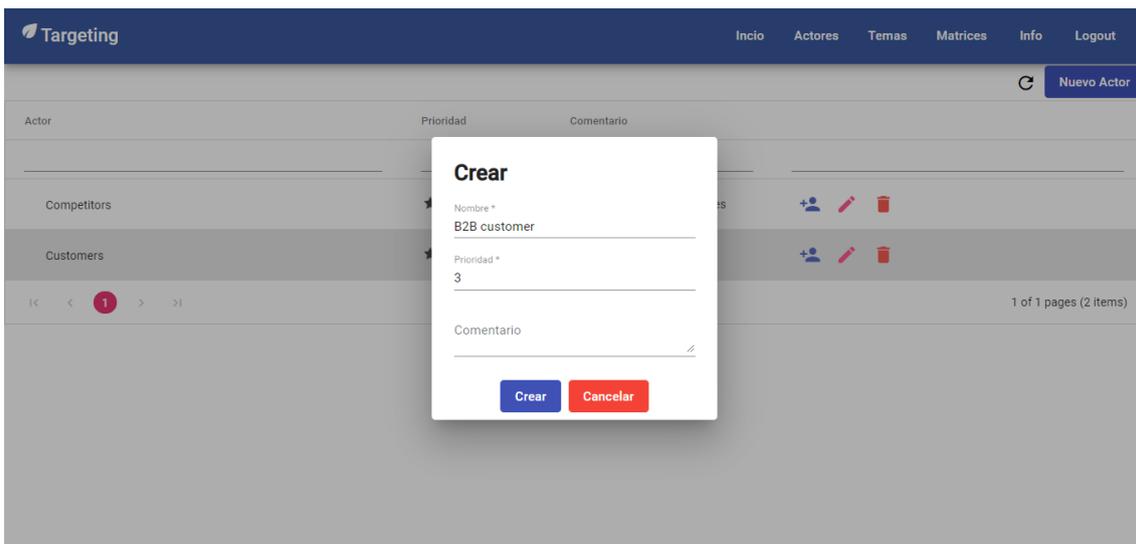
Figura 36. Comprobación de la creación de un actor tipo padre. Elaborado por el autor.

Se desplegará el formulario para la creación del actor tipo sub-actor que se puede visualizar en la Figura 37.

Aquí se podrá ingresar los siguientes datos:

- Nombre: Tipo Texto
- Prioridad: Numero del 1 al 3
- Comentario: Tipo Texto o vacío

Una vez ingresado los datos se procederá a pulsar el botón “**Crear**”.



Actor	Prioridad	Comentario	
Competitors	★ ★ ★		  
Customers	★ ★ ★		  

Figura 37. Formulario de Creación de Actor tipo sub-actor de la aplicación web. Elaborado por el autor.

Una vez creado el sub-actor se mostrará en la lista de actores que se puede visualizar en la Figura 38. Se repetirá el proceso para los actores y sub-actores que se desee crear.

Actor	Prioridad	Comentario	
Competitors	★ ★ ★	Los competidores principales	+ ✎ 🗑
▼ Customers	★ ★ ★		+ ✎ 🗑
↳ B2B customer	★ ★ ★		+ ✎ 🗑

Figura 38. Comprobación de la creación de un actor tipo sub-actor. Elaborado por el autor.

Una vez creado todos los actores y sub-actores necesarios, se obtendrá una lista como la que se puede visualizar en la Figura 39.

Actor	Prioridad	Comentario	
Competitors	★ ★ ★	Los competidores principales	+ ✎ 🗑
▼ Customers	★ ★ ★		+ ✎ 🗑
↳ B2B customer	★ ★ ★		+ ✎ 🗑
↳ Customers of our customers	★ ★ ★		+ ✎ 🗑
▼ Suppliers	★ ★ ★		+ ✎ 🗑
↳ Tier 1	★ ★ ★		+ ✎ 🗑
↳ Tier 2	★ ☆ ☆		+ ✎ 🗑
↳ Tier 3	★ ★ ★		+ ✎ 🗑

Figura 39. Ejemplo de una lista completa de actores. Elaborada por el autor.

Existe también la función de eliminar que está representado por el ícono del basurero en el lado derecho de cada actor. Si se pulsa el ícono para un actor específico se mostrará el mensaje de confirmación para eliminar al actor, que se puede visualizar en la Figura 40.

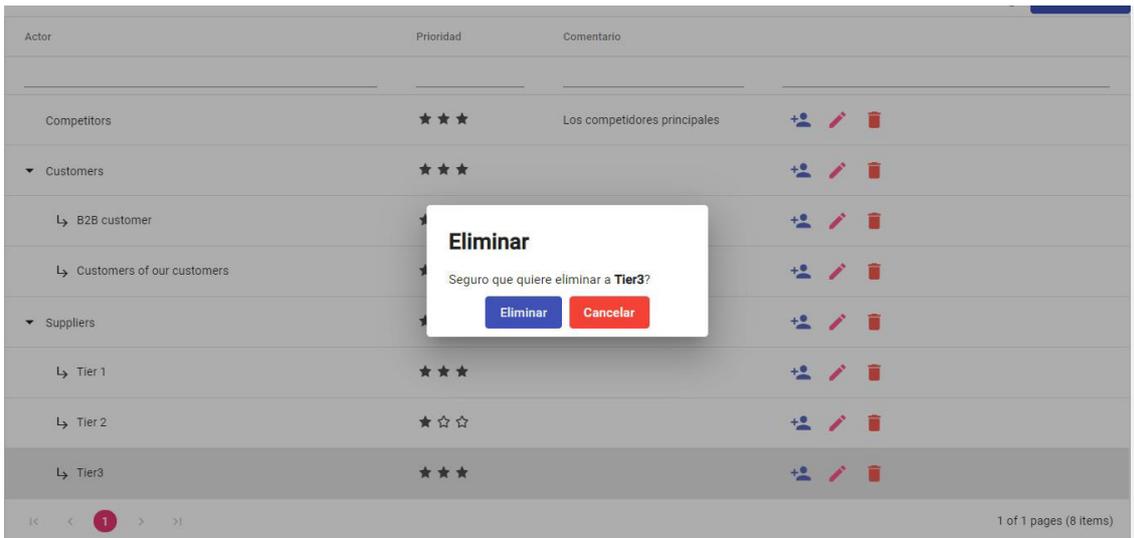


Figura 40. Mensaje de confirmación para eliminar actor de la aplicación web. Elaborado por el autor.

También se puede editar un actor mediante el botón que está representado por el ícono del lápiz en el lado derecho de cada actor. Si se pulsa el ícono para un actor específico se mostrará el formulario de edición para editar al actor, que se puede visualizar en la Figura 41.

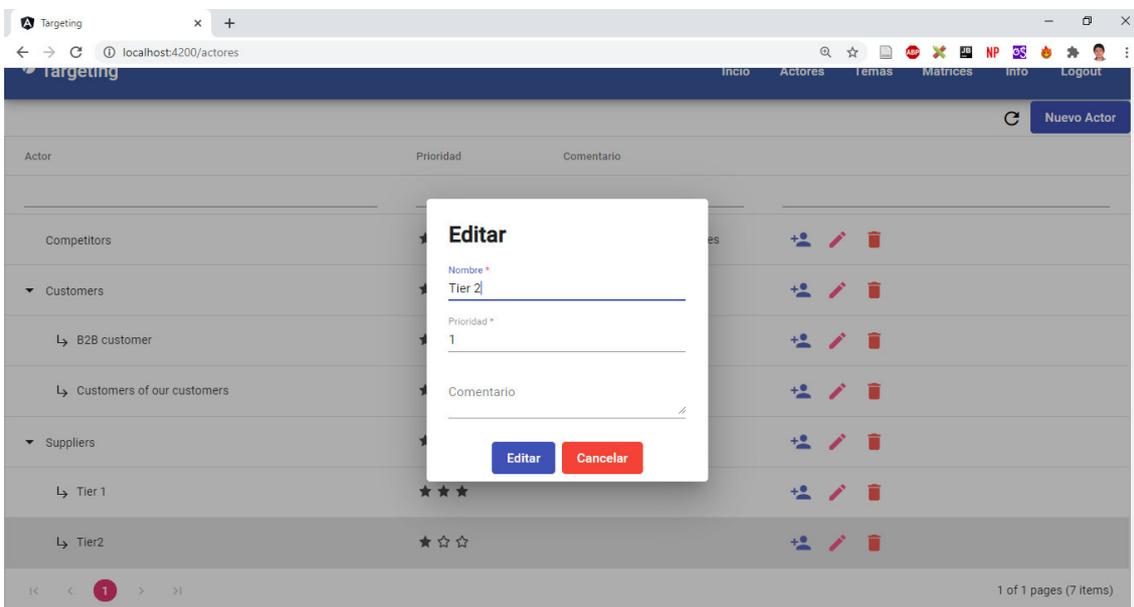


Figura 41. Formulario de edición de un actor de la aplicación web. Elaborado por el autor.

Se puede también filtrar los actores mediante una función que se puede acceder escribiendo el nombre de actor o buscando el número de prioridad de cierto

actor en los campos que están en la parte superior de la lista. Un ejemplo de este filtro se puede visualizar en la Figura 42.

Actor	Prioridad	Comentario
Tier 1		
Suppliers	★★★	
Tier 1	★★★	

1 of 1 pages (2 items)

Figura 42. Ejemplo del filtro por actor de la aplicación web. Elaborado por el autor.

Pantalla de Temas

Como paso inicial se necesitará definir temas principales que puedan o no tener sub-temas debajo de ellos. Primero pulsamos el botón de temas, sea en la pantalla inicial o la barra de navegación. En la Figura 43 se puede visualizar la página de temas.

Tema	Prioridad	Comentario
No records to display		

0 of 0 pages (0 items)

Figura 43. Página de Temas de la aplicación web. Elaborado por el autor.

En esta pantalla primero se tendrán que crear los temas padres que son los temas con números en la lista anteriormente creada. Para crear un padre se necesita pulsar el botón que se encuentra en la parte superior izquierda de la pantalla llamado “Nuevo Tema”. Una vez esto se desplegará el formulario de creación que se visualiza en la Figura 44.

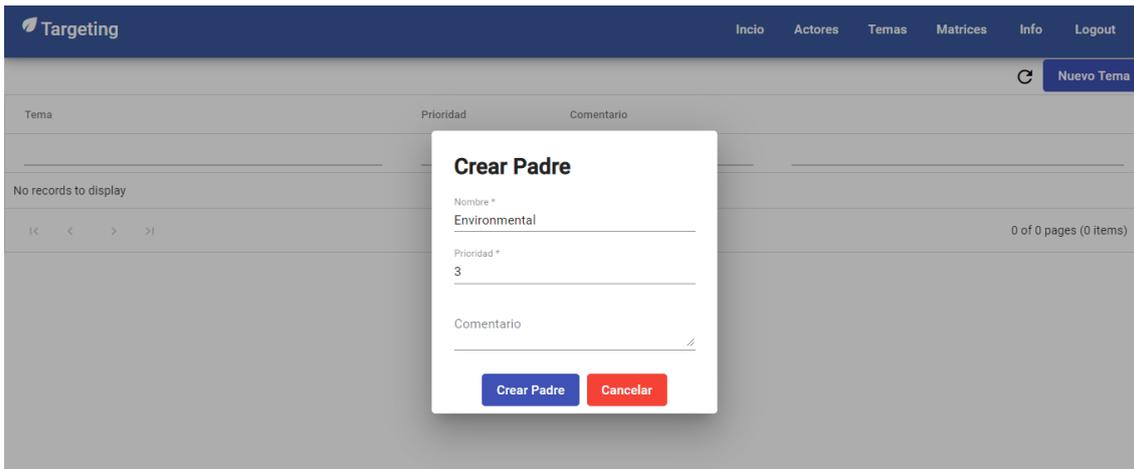


Figura 44. Formulario de Creación de Tema tipo Padre de la aplicación web. Elaborado por el autor.

Aquí se podrán ingresar los siguientes datos:

- Nombre: Tipo Texto
- Prioridad: Numero del 1 al 3
- Comentario: Tipo Texto o vacío

Una vez ingresados los datos se procederá a pulsar el botón “Crear Padre”. Una vez creado el tema se mostrará en la lista de actores que se puede visualizar en la Figura 45. Por ejemplo, para crear un sub-tema dentro del tema “Environmental” se pulsará el botón  en la fila correspondiente al actor escogido.

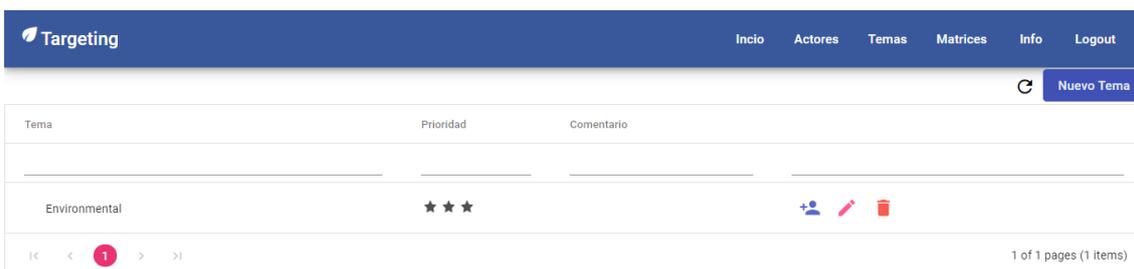


Figura 45. Comprobación de la creación de un tema tipo padre. Elaborado por el autor.

A continuación, se desplegará el formulario para la creación del tema tipo sub-tema que se puede visualizar en la figura 46.

Aquí se podrá ingresar los siguientes datos:

- Nombre: Tipo Texto
- Prioridad: Numero del 1 al 3
- Comentario: Tipo Texto o vacío

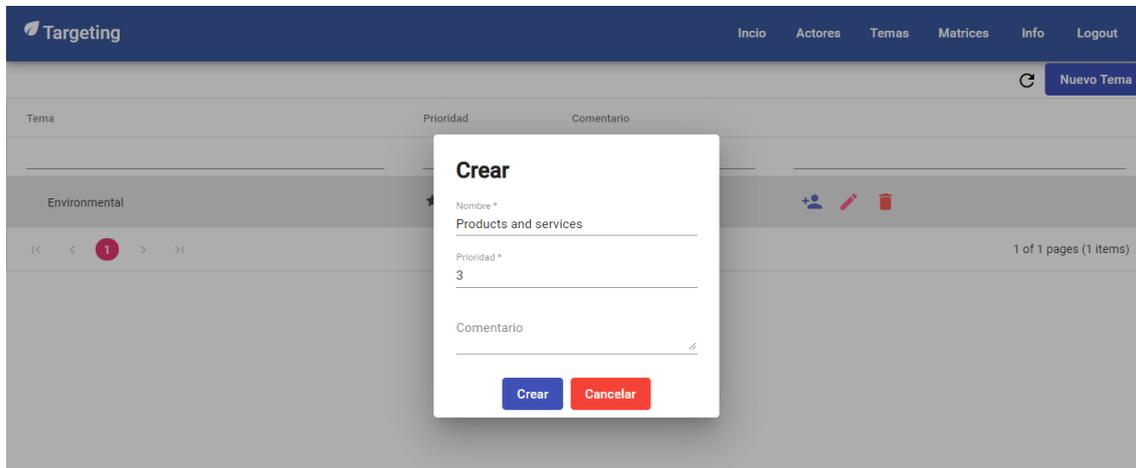
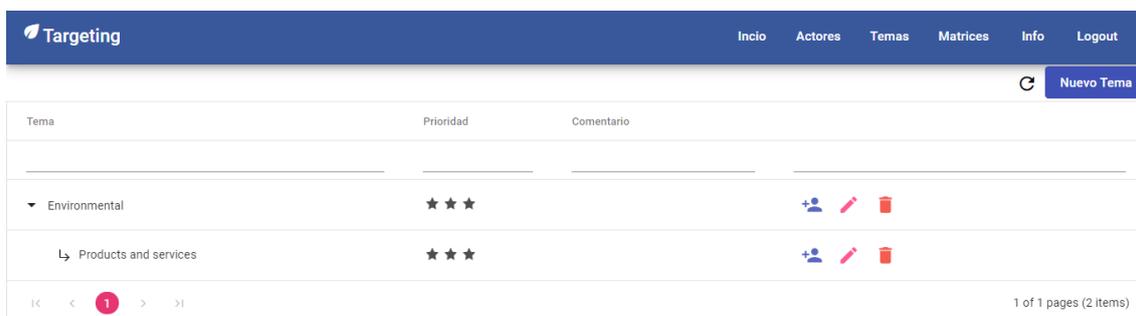


Figura 46. Formulario de Creación de Tema tipo sub-tema de la aplicación web. Elaborado por el autor.

Una vez ingresado los datos se procederá a pulsar el botón “**Crear**”. Cuando se haya creado el tema, éste se mostrará en la lista de actores que se puede visualizar en la Figura 47. Se repetirá el proceso para los temas y sub-temas que se desee crear.



Tema	Prioridad	Comentario
Environmental	★★★	
↳ Products and services	★★★	

Figura 47. Comprobación de la creación de un actor tipo sub-tema. Elaborado por el autor.

Una vez creados todos los temas y sub-temas necesarios, se obtendrá una lista como la que se puede visualizar en la Figura 48.

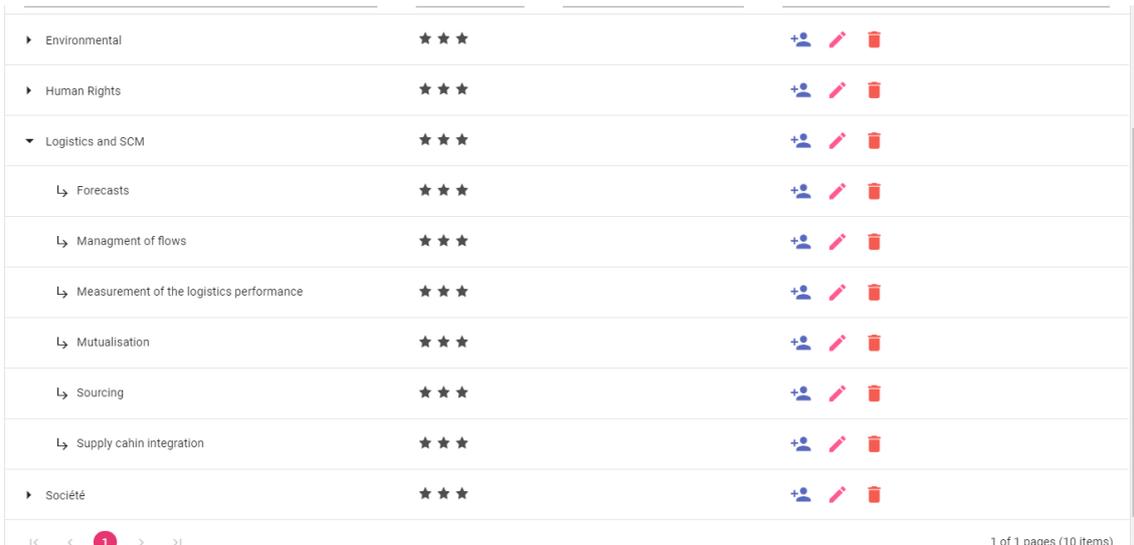


Figura 48. Ejemplo de una lista completa de temas. Elaborada por el autor.

Existe la funci n de eliminar que est  representado por el icono del basurero en el lado derecho de cada tema. Si se pulsa el icono para un tema espec fico se mostrar  el mensaje de confirmaci n para eliminar al tema, que se puede visualizar en la Figura 49.

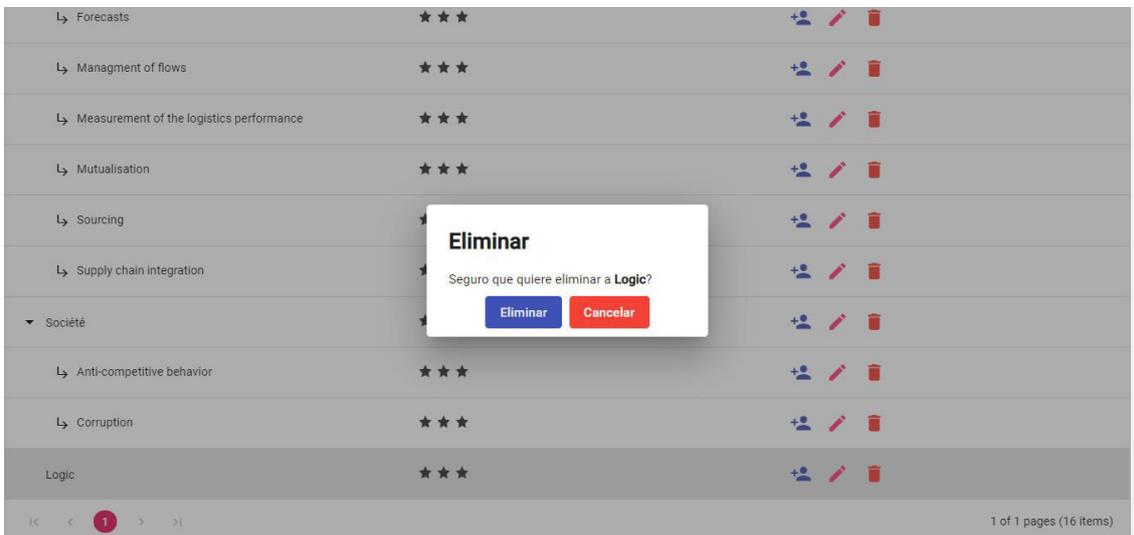


Figura 49. Mensaje de confirmaci n para eliminar tema de la aplicaci n web. Elaborado por el autor.

Existe tambi n la funci n de editar que est  representado por el icono del l piz en el lado derecho de cada tema. Si se pulsa el icono para un tema espec fico

se mostrará el formulario de edición para editar al tema, que se puede visualizar en la Figura 50.

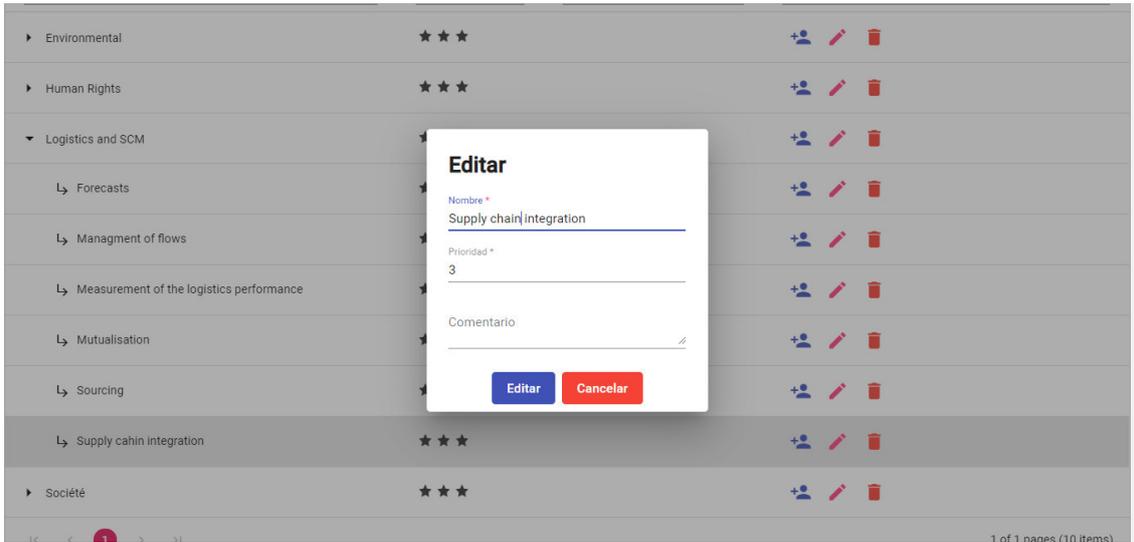


Figura 50. Formulario de edición de un tema de la aplicación web. Elaborado por el autor.

Finalmente, también existe la función de filtrar que se puede acceder escribiendo el nombre de tema o buscando el número de prioridad de cierto tema en los campos que están en la parte superior de la lista. Un ejemplo de este filtro se puede visualizar en la Figura 51.

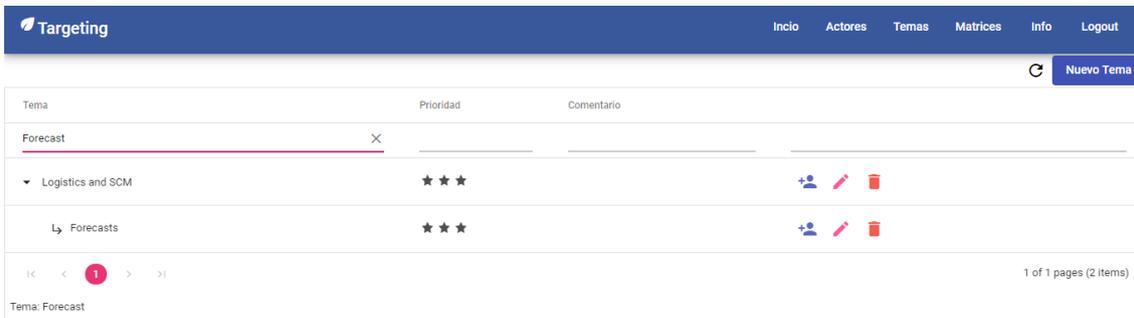


Figura 51. Ejemplo del filtro por tema de la aplicación web. Elaborado por el autor.

Pantalla de Matrices

Finalmente, para concluir con el proceso de Targeting, se creará una matriz de actores y temas. Para ello, primero pulsamos el botón de matrices, sea en la pantalla inicial o la barra de navegación. En la Figura 52 se puede visualizar la página de matrices.

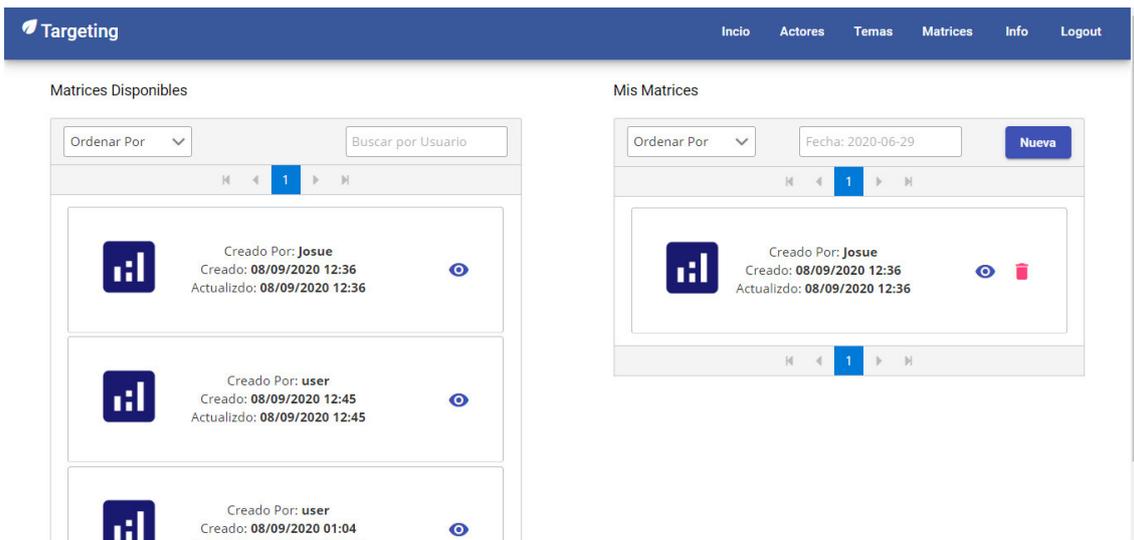


Figura 52. Página de Matrices de la aplicación web. Elaborado por el autor.

En esta pantalla primero se tendrá que crear una nueva matriz. Para crear una matriz tenemos que pulsar el botón que se encuentra en la parte superior derecha de la pantalla llamado “Nuevo”. Una vez hecho esto, se desplegará un mensaje de confirmación que se puede visualizar en la Figura 53. Si el usuario pulsa en “Sí”, el sistema lo dirigirá a la página de matriz que se puede visualizar en la Figura 55.

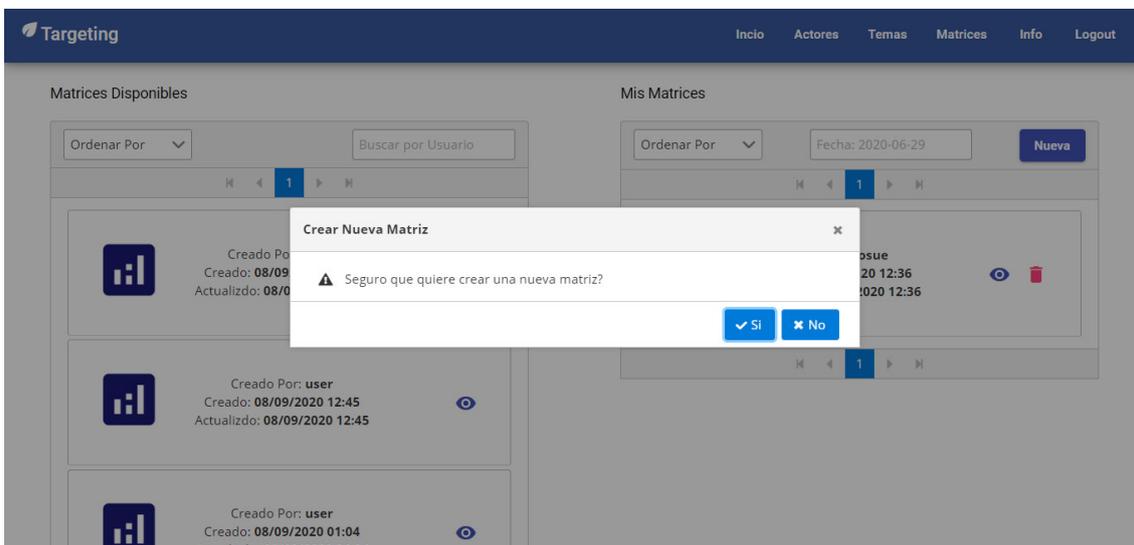


Figura 53. Mensaje de confirmación de creación de matriz de la aplicación web. Elaborado por el autor.

Existe también la función de eliminar que está representada por el ícono del basurero en el lado derecho de cada matriz que le pertenece al usuario que inicio la sesión. Si se pulsa el ícono para una matriz específica se mostrará el mensaje de confirmación para eliminar a la matriz (Figura 54).

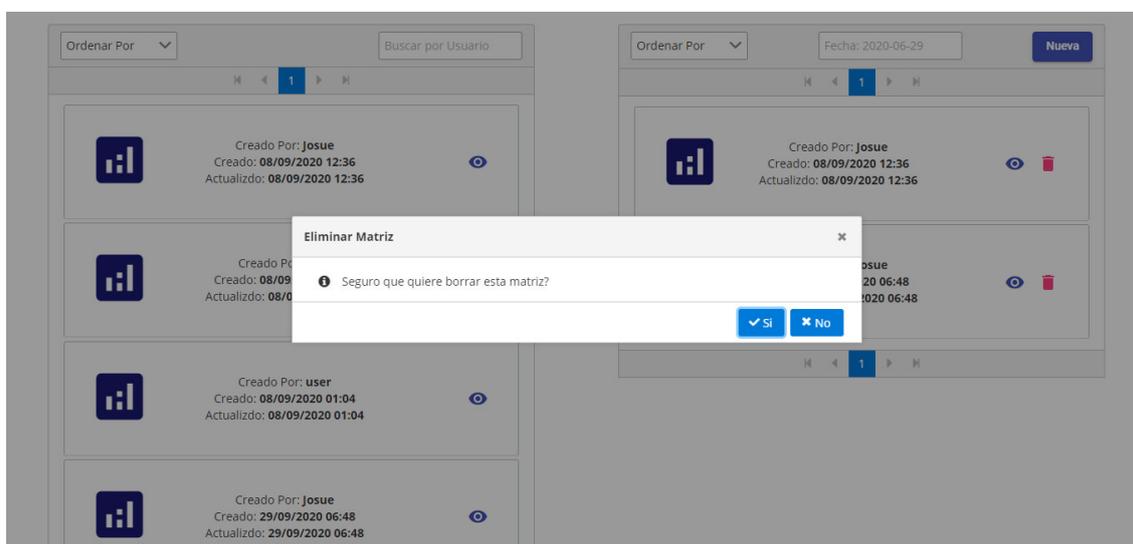


Figura 54. Mensaje de confirmación para eliminar matriz de la aplicación web.
Elaborado por el autor.

También existe la función de visualizar una matriz que está representado por el ícono del ojo en el lado derecho de cada matriz. Si se pulsa el ícono para una matriz específica, se redirigirá al usuario a la pantalla de esa matriz específica, que se puede visualizar en la Figura 55.

Pantalla de Matriz

En esta pantalla se podrá visualizar la matriz que se escoja de la pantalla anterior. Esta matriz contendrá a los actores, temas creados, y las celdas para representar los cruces entre ellos que se puede visualizar en la Figura 55.

Temas/Actores			Competitors		Customers				Suppliers			
Environmental	Products and services	Eco-design	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
Human Rights	Investments and procurement practices	Eco-design	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
			N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
	Forecasts	Eco-design	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
			N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
	Management of flows	Eco-design	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
			N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●

Figura 55. Página de Matriz de la aplicación web. Elaborado por el autor.

En esta matriz se puede dar una prioridad y un tiempo a los cruces de Actor y Temas que se sean relevantes. También, en la parte superior de la matriz existe un filtro de prioridad de tema y actor cuyo uso se puede visualizar en la Figura 56 y la Figura 57.

Temas/Actores			Competitors		Customers				Suppliers			
Environmental	Products and services	Eco-design	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
Human Rights	Investments and procurement practices	Eco-design	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
			N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
	Forecasts	Eco-design	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
			N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
	Management of flows	Eco-design	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●
			N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●	N/A	● ● ● ●

Figura 56. Filtro de Actor por prioridad de la aplicación web. Elaborado por el autor.

Temas/Actores			Competitors	Customers			Tier 1	Suppliers
Environmental	Products and services	Eco-design	N/A	●	●	●	●	●
			N/A	🕒	🕒	🕒	🕒	🕒
Human Rights	Investments and procurement practices		N/A	●	●	●	●	●
			N/A	🕒	🕒	🕒	🕒	🕒
Logistics and SCM	Forecasts		N/A	●	●	●	●	●
			N/A	🕒	🕒	🕒	🕒	🕒
	Management of flows		N/A	●	●	●	●	●
			N/A	🕒	🕒	🕒	🕒	🕒
	Measurement of the logistics performance		N/A	●	●	●	●	●
			N/A	🕒	🕒	🕒	🕒	🕒

Figura 57. Filtro de Tema por prioridad de la aplicación web. Elaborado por el autor.

Una vez puesto los filtros, las filas y columnas que cumplen esta condición se pintarán de color amarillo como se puede visualizar en la Figura 58.

Temas/Actores			Competitors	Customers			Suppliers	
Environmental	Products and services	Eco-design	N/A	●	●	●	●	●
			N/A	🕒	🕒	🕒	🕒	🕒
Human Rights	Investments and procurement practices		N/A	●	●	●	●	●
			N/A	🕒	🕒	🕒	🕒	🕒
Logistics and SCM	Forecasts		N/A	●	●	●	●	●
			N/A	🕒	🕒	🕒	🕒	🕒
	Management of flows		N/A	●	●	●	●	●
			N/A	🕒	🕒	🕒	🕒	🕒
	Measurement of the logistics performance		N/A	●	●	●	●	●
			N/A	🕒	🕒	🕒	🕒	🕒

Figura 58. Filtro por prioridad de actor y tema en la aplicación web. Elaborado por el autor.

Cuando el proceso de targeting y la discusión con los colaboradores se haya completado, se obtendrá una matriz como la que se puede visualizar en la Figura 59.

Temas/Actores			Competitors	Customers				Suppliers									
				B2B customer		Customers of our customers		Tier 1		Tier 2							
Environmental	Products and services	Eco-design	N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●
			N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●
Human Rights	Investments and procurement practices		N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●
			N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●
Logistics and SCM	Forecasts		N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●
			N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●
	Management of flows		N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●
			N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●
	Measurement of the logistics performance		N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●
			N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●
	Mutualisation		N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●
			N/A	●	●	●	●	N/A	●	●	●	●	N/A	●	●	●	●

Figura 59. Ejemplo de una matriz de targeting completa. Elaborado por el autor.

3.2 Resultados de la Evaluación de Aceptación

Se utilizó una evaluación cualitativa capaz de medir la aceptación del sistema. Para poder obtener estos resultados, se utilizó una encuesta que fue entregada a cada uno de los participantes de las pruebas de aceptación del sistema. La encuesta utilizó preguntas directamente relacionadas con los parámetros en los que basa TAM. Además, una pregunta final fue destinada a reunir observaciones o recomendaciones por parte de los participantes. Para poder optimizar y facilitar la distribución de la encuesta, se usó Google Forms (Figura 60).

¿Considera que la herramienta utilizada es fácil de usar? *

Long answer text

Tomando en cuenta el objetivo de facilitar el metodo de Targeting. ¿Considera usted que esta herramienta es útil? *

Long answer text

¿Tiene alguna recomendación de mejora? *

Long answer text

Figura 60. Modelo de encuesta. Elaborado por el autor.

Los participantes de las pruebas fueron ocho alumnos de la carrera de sistemas que están en los últimos semestres de la carrera y que están trabajando en temas de titulación relacionados con SScan. Para poder realizar estas pruebas se necesitó un grupo de personas que tenga un conocimiento básico sobre el SScan. Ya que este sistema automatiza el segundo proceso en el workflow del SScan y puede llegar a ser confuso el objetivo y enfoque de la aplicación para las personas que no tienen ningún contexto o conocimiento previo de él. Antes de realizar esta prueba se realizó una prueba preliminar con el PhD. Edison Loza Aguirre para su aprobación. Una vez recibida la misma, se coordinó una reunión con los participantes de las pruebas. Ya que este sistema está enfocado para una interacción colaborativa se decidió dividir a los participantes en varios grupos y simular una reunión de targeting. Así, se dividió a los participantes en dos grupos de tres personas y un grupo de dos personas. Las pruebas concluyeron sin mayor inconveniente y los participantes llenaron la encuesta que proporcionaron los siguientes resultados.

Facilidad de uso: De 8 personas, 7 (88%) consideraron que el sistema es fácil de usar (Figura 61).



Figura 61. Resultados facilidad de uso del sistema. Elaborado por el autor.

Algunos de los comentarios recibidos en la facilidad de uso fueron:

- “De acuerdo con lo mencionado por la metodología TAM, creo que la herramienta es intuitiva y fácil de utilizar.”
- “Si es intuitiva y no requiere mucho esfuerzo”

Utilidad percibida: Las ocho (100%) personas consideran que el sistema es de utilidad para la tarea colaborativa de Targeting (Figura 62).

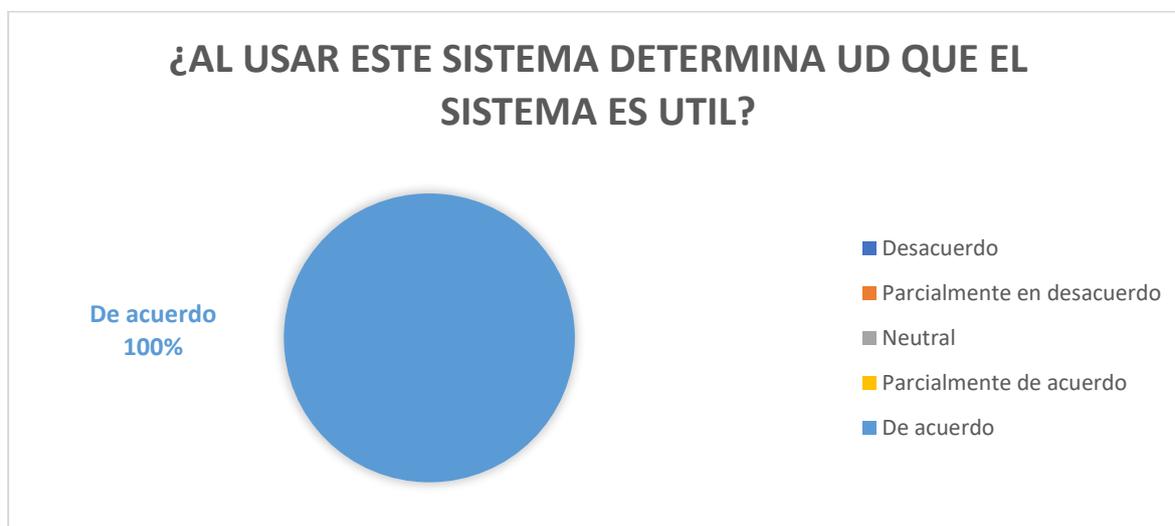


Figura 62. Resultados utilidad percibida del sistema. Elaborado por el autor.

Algunos de comentarios recibidos fueron:

- “La herramienta es útil, especialmente para reuniones en tiempo real entre varios usuarios”
- “la herramienta presenta una tabla con valiosa información y con trabajo en tiempo real”

Algunas de las recomendaciones que proveyeron los participantes y que se consideran las más relevantes y acertadas fueron:

- “Al crear una matriz grande, se debería buscar una opción para que los temas y actores permanezcan estáticos y seguir navegando en la matriz.”
- “En las pantallas de actor y tema, sería bueno poner un placeholder para identificar las búsquedas”
- “Respecto a la autenticación tener más controles para crear usuarios. Respecto a la accesibilidad mensajes de apoyo o ayuda de controles.”
- “Se puede mejorar el diseño gráfico de la app web”

4 CONCLUSIONES

- El proceso de Targeting de SScan, usual y tradicionalmente, es realizado con la ayuda de las herramientas “manuales” como son un pizarrón y marcadores. El sistema desarrollado en el presente trabajo logra transmitir este proceso a una plataforma digital, facilitando así la definición de actores y temas y la creación de la matriz de targeting.
- La funcionalidad implementada en el sistema significa un paso importante en el proceso de SScan, así como también en proceso de targeting. Según sugieren los resultados, los usuarios pudieron crear, visualizar, y utilizar el sistema con facilidad.
- Los resultados de evaluación del sistema demuestran, de manera general, que los artefactos desarrollados en el presente trabajo permiten apoyar en el proceso colaborativo del proceso de Targeting que una organización pueda llegar a ejecutar, impactando de manera positiva a otros procesos relacionados como la toma de decisiones, el procesamiento de datos, etc.
- Todo sistema debe tomar en cuenta el principio de usabilidad con el fin que los usuarios no tengan complicaciones y que puedan usarlo de una manera fácil e intuitiva. De acuerdo con los resultados de la prueba se puede visualizar que existe una aceptación completa del 88% y una aceptación parcial del 12% de parte de los participantes de las pruebas que indican que el sistema es fácil de usar.
- La integración de las herramientas de GraphQL y Prisma fueron clave para la creación del sistema ya que facilitaron las consulta y modificación de datos. GraphQL facilita las consultas a la base de datos y brindó una estructura de objeto a los datos recuperados de la base de datos. Esto ayuda a facilitar la manipulación de los datos en el sistema y también ayuda a reducir el tiempo que se gastaría en la creación de consultas normales de base de datos para conseguir ciertos datos relacionados a varias entidades.
- El uso de una metodología de prototipado aportó mucho al desarrollo del sistema. Esta metodología ayudó a recibir retroalimentación rápida y

concreta para enfocarse en el mejoramiento del sistema para el próximo prototipo. Esta metodología facilita a la organización y elaboración del sistema ya que el grupo de trabajo solo era formado por una persona, y por esta razón se optó por el uso de esta metodología en vez de una metodología como Scrum.

5 RECOMENDACIONES

- Por las circunstancias que está pasando el mundo con la situación del COVID-19 se podría integrar una forma de comunicación virtual en el sistema como por ejemplo un chat o hasta un componente para video llamadas integradas.
- Se recomendaría implementar un video tutorial del funcionamiento, un “tutorial guiado”, o mensajes de ayuda a través de la aplicación para que los usuarios con poca experiencia en el uso de páginas web puedan obtener el máximo beneficio del uso del sistema.
- Se podría hacer una división por tópicos para así facilitar la organización de temas y actores, y así generar una clasificación y búsqueda más optimizada y ordena. También incluir las fuentes para cada actor y tema ya que esto puede ser de gran importancia para los usuarios finales al momento de clasificar prioridades, ya que una fuente confiable puede tener más impacto que una fuente amateur en la toma de decisiones.
- El sistema fue desarrollado para un uso fácil e intuitivo para el usuario, pero existen ciertos elementos visuales de las interfaces que se podrían mejorar aún. Se necesitaría invertir un poco más de tiempo en la parte de diseño y colores para que éste tenga una vista más pulida y sea totalmente aceptable por parte del usuario.

BIBLIOGRAFÍA

- [1] O. El, "Personal Information Systems for Strategic Scanning in Turbulent Environments: Can the CEO Go On-Line?", *Manag. Inf. Syst. Q.*, vol. 9, n° 1, dic. 1985, [En línea]. Disponible en: <https://aisel.aisnet.org/misq/vol9/iss1/4>
- [2] E. Loza-Aguirre, N. Lesca, H. Haddad, y M.-L. Caron-Fasan, "Getting a Clean Shot on a Blurred Target: Improving Targeting for Strategic Scanning through Action Research in 10 French Organizations", *Commun. Assoc. Inf. Syst.*, vol. 39, n° 1, dic. 2016, doi: 10.17705/1CAIS.03927.
- [3] G. Elofson, "Using AI-based monitoring technologies in management control", en *[1993] Proceedings of the Twenty-sixth Hawaii International Conference on System Sciences*, ene. 1993, vol. iv, pp. 301–310 vol.4. doi: 10.1109/HICSS.1993.284198.
- [4] H. Chen, M. Chau, y D. Zeng, "CI Spider: a tool for competitive intelligence on the Web", *Decis. Support Syst.*, vol. 34, n° 1, pp. 1–17, dic. 2002, doi: 10.1016/S0167-9236(02)00002-7.
- [5] Y. Zhang, Y. Dang, H. Chen, M. Thurmond, y C. Larson, "Automatic online news monitoring and classification for syndromic surveillance", *Decis. Support Syst.*, vol. 47, n° 4, pp. 508–517, nov. 2009, doi: 10.1016/j.dss.2009.04.016.
- [6] M. Saravanan y A. Das, "Smart real-time meeting room", en *2017 IEEE Region 10 Symposium (TENSYMP)*, jul. 2017, pp. 1–5. doi: 10.1109/TENCONSpring.2017.8070069.
- [7] N. Lesca y M.-L. Caron-Fasan, "Strategic Scanning Project Failure and Abandonment Factors: Lessons Learned", *Eur. J. Inf. Syst.*, vol. 17, pp. 371–386, sep. 2008, doi: 10.1057/ejis.2008.21.
- [8] E. Loza-Aguirre y A. Buitrago Hurtado, "Qualitative Assessment of User Acceptance within Action Design Research and Action Research: Two Case Studies", *Lat. Am. J. Comput.*, vol. 1, pp. 7–16, oct. 2014.
- [9] W. Chung, "BizPro: Extracting and categorizing business intelligence factors from textual news articles", *Int. J. Inf. Manag.*, vol. 34, n° 2, pp. 272–284, abr. 2014, doi: 10.1016/j.ijinfomgt.2014.01.001.
- [10] "Strategic Decisions and Weak Signals: Anticipation for Decision-Making | Wiley", *Wiley.com*. <https://www.wiley.com/en-ec/Strategic+Decisions+and+Weak+Signals%3A+Anticipation+for+Decision+Making-p-9781848216099> (accedido jul. 06, 2020).
- [11] C. J. Torrecilla-Salinas, J. Sedeño, M. J. Escalona, y M. Mejías, "Estimating, planning and managing Agile Web development projects under a value-based perspective", *Inf. Softw. Technol.*, vol. 61, pp. 124–144, may 2015, doi: 10.1016/j.infsof.2015.01.006.
- [12] H. M. Abdullah y A. M. Zeki, "Frontend and Backend Web Technologies in Social Networking Sites: Facebook as an Example", en *2014 3rd International Conference on Advanced Computer Science Applications and Technologies*, Amman, Jordan, dic. 2014, pp. 85–89. doi: 10.1109/ACSAT.2014.22.
- [13] "The Apollo platform", *Apollo GraphQL Docs*. <https://www.apollographql.com/docs/intro/platform/> (accedido jul. 06, 2020).

- [14] “Introduction”, *Apollo GraphQL Docs*.
<https://www.apollographql.com/docs/react/> (accedido jul. 06, 2020).
- [15] “Docker overview”, *Docker Documentation*, jul. 03, 2020.
<https://docs.docker.com/get-started/overview/> (accedido jul. 06, 2020).
- [16] “GraphQL: A query language for APIs.” <http://graphql.org/> (accedido jul. 06, 2020).
- [17] “Software de diagramación en línea y solución visual”, *Lucidchart*.
<https://www.lucidchart.com> (accedido sep. 14, 2020).
- [18] “Centro de aprendizaje”, *Lucidchart*.
<https://www.lucidchart.com/pages/es/recorrido-por-el-producto> (accedido sep. 14, 2020).
- [19] “PostgreSQL: About”. <https://www.postgresql.org/about/> (accedido jul. 06, 2020).
- [20] “Prisma 1.34 - Prisma Introduction: What, Why & How”, *Prisma*.
<https://v1.prisma.io/docs/1.34/understand-prisma/prisma-introduction-what-why-how-j9ff> (accedido jul. 06, 2020).
- [21] “Crea y comparte diseños, maquetas y prototipos rápidamente | Adobe XD”. <https://www.adobe.com/la/products/xd/details.html> (accedido sep. 14, 2020).
- [22] “Angular - Introduction to Angular concepts”.
<https://angular.io/guide/architecture> (accedido jul. 07, 2020).
- [23] A. C. Team, “Angular Material”, *Angular Material*.
<https://material.angular.io/> (accedido jul. 07, 2020).
- [24] M. O. contributors Jacob Thornton, and Bootstrap, “Introduction”.
<https://getbootstrap.com/docs/4.5/getting-started/introduction/> (accedido sep. 14, 2020).
- [25] “Git”. <https://git-scm.com/> (accedido sep. 14, 2020).
- [26] “Build software better, together”, *GitHub*. <https://github.com> (accedido sep. 14, 2020).
- [27] “Index | Node.js v14.5.0 Documentation”. <https://nodejs.org/api/> (accedido jul. 07, 2020).
- [28] “About npm | npm Documentation”. <https://docs.npmjs.com/about-npm/> (accedido sep. 14, 2020).
- [29] “PrimeNG”. <https://www.primefaces.org/primeng/showcase/#/> (accedido sep. 14, 2020).
- [30] “Documentation · TypeScript”.
<https://www.typescriptlang.org/docs/home.html> (accedido jul. 07, 2020).
- [31] “Visual Studio Code Frequently Asked Questions”.
<https://code.visualstudio.com/docs/supporting/faq> (accedido jul. 07, 2020).
- [32] “Why Visual Studio Code?”
<https://code.visualstudio.com/docs/editor/whyvscode> (accedido jul. 07, 2020).
- [33] V. S. Gordon y J. M. Bieman, “Rapid prototyping: lessons learned”, *IEEE Softw.*, vol. 12, nº 1, pp. 85–95, ene. 1995, doi: 10.1109/52.363162.
- [34] I. Sommerville, *Software Engineering*. 2015, pp.44-47.
- [35] R. S. Pressman, *Software Engineering: A Practitioner’s Approach*. McGraw-Hill Education, 2010, pp.43-45.
- [36] A. Coyette, S. Kieffer, y J. Vanderdonckt, “Multi-fidelity Prototyping of User Interfaces”, en *Human-Computer Interaction – INTERACT 2007*, Berlin, Heidelberg, 2007, pp. 150–164. doi: 10.1007/978-3-540-74796-3_16.

- [37] W. R. Bischofberger y G. Pomberger, *Prototyping-Oriented Software Development: Concepts and Tools*. Berlin Heidelberg: Springer-Verlag, 1992. doi: 10.1007/978-3-642-84760-8.
- [38] D. A. Fern y S. E. Donaldson, "Tri-Cycle: a prototype methodology for advanced software development", en *[1989] Proceedings of the Twenty-Second Annual Hawaii International Conference on System Sciences. Volume II: Software Track*, ene. 1989, vol. 2, pp. 377–386 vol.2. doi: 10.1109/HICSS.1989.48015.
- [39] K. K. Breitman, J. C. S. do P. Leite, J. Cesar, y S. P. Leite, *Managing User Stories*.
- [40] R. Agarwal y D. Umphress, "Extreme programming for a single person team", en *Proceedings of the 46th Annual Southeast Regional Conference on XX*, New York, NY, USA, mar. 2008, pp. 82–87. doi: 10.1145/1593105.1593127.
- [41] K. Mathieson, E. Peacock, y W. W. Chin, "Extending the technology acceptance model: the influence of perceived user resources", *ACM SIGMIS Database*, vol. 32, nº 3, p. 86, jul. 2001, doi: 10.1145/506724.506730.
- [42] Y. Lee, K. A. Kozar, y K. R. T. Larsen, "The Technology Acceptance Model: Past, Present, and Future", *Commun. Assoc. Inf. Syst.*, vol. 12, 2003, doi: 10.17705/1CAIS.01250.
- [43] F. D. Davis, "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology", *MIS Q.*, vol. 13, nº 3, pp. 319–340, 1989, doi: 10.2307/249008.
- [44] A. L. Lederer, D. J. Maupin, M. P. Sena, y Y. Zhuang, "The technology acceptance model and the World Wide Web", *Decis. Support Syst.*, vol. 29, nº 3, pp. 269–282, oct. 2000, doi: 10.1016/S0167-9236(00)00076-2.
- [45] L. G. Wallace y S. D. Sheetz, "The adoption of software measures: A technology acceptance model (TAM) perspective", *Inf. Manage.*, vol. 51, nº 2, pp. 249–259, mar. 2014, doi: 10.1016/j.im.2013.12.003.
- [46] "Subscriptions", *Apollo GraphQL Docs*. <https://www.apollographql.com/docs/react/data/subscriptions/> (accedido sep. 28, 2020).
- [47] C. Marion y J. Jomier, "Real-time collaborative scientific WebGL visualization with WebSocket", en *Proceedings of the 17th International Conference on 3D Web Technology - Web3D '12*, Los Angeles, California, 2012, p. 47. doi: 10.1145/2338714.2338721.

GLOSARIO

Vigilancia Estratégica: Proceso que consiste en recolectar y procesar información externa con el fin de entender a los actores de cambio que pueden influenciar directa e indirectamente sobre el negocio de dicha organización.

SScan: Acrónimo de Vigilancia Estratégica.

Aplicación Web: Programa o aplicación que está almacenado en un servidor remoto y al que se accede generalmente mediante Internet.

GraphQL: Lenguaje de consulta para APIs desarrollado por Facebook. Representa una alternativa a REST.

Javascript: Lenguaje dinámico e interpretado diseñado principalmente para ser ejecutado en navegadores web y actualmente en servidores web.

Typescript: Extensión de Javascript que añade características útiles como un sistema de tipos.

Node.js: Es un ambiente de ejecución asíncrono y dirigido por eventos para programas escritos en Javascript.

PostgreSQL: Sistema de gestión de bases de datos relacionales con enfoque en extensibilidad y cumplimiento de estándares técnicos SQL.

Docker: Tecnología de virtualización de contenedores enfocados en el bajo consumo de recursos.

TAM: Acrónimo de Technology Acceptance Model (o Modelo de Aceptación de Tecnología). Es una teoría usada en Sistemas de Información que modela el cómo los usuarios pueden llegar a aceptar o usar una determinada tecnología.