

Universidad ORT Uruguay
Facultad de Ingeniería

**TechBuildMe: plataforma web de aprendizaje
para el rubro de la construcción**

Entregado como requisito para la obtención del título de Ingeniero de Software

Martina Severo – 192680

Santiago Tonarelli – 229484

Tutora: Mariana Lasarte

2023

Declaración de autoría

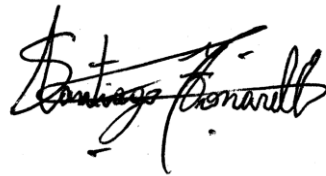
Nosotros, Martina Severo y Santiago Tonarelli, declaramos que el trabajo que se presenta en esta obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el proyecto de grado de la carrera de Ingeniería de Software;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



Martina Severo

30 de marzo de 2023



Santiago Tonarelli

30 de marzo de 2023

Agradecimientos

En primer lugar, queremos agradecer a nuestros familiares y amigos por su apoyo incondicional a lo largo de estos años. Su presencia, aliento y cariño han sido esenciales para alcanzar este logro.

En segundo lugar, queremos hacer una mención especial a nuestra tutora, Mariana Lasarte, quien nos acompañó en todo momento, brindándonos su experiencia, consejos y orientación, incluso en los momentos más difíciles del proyecto.

Asimismo, darles las gracias a todas las personas que generosamente nos brindaron su tiempo y colaboración para validar nuestra idea. En particular, queremos destacar el apoyo de la Arq. Fernanda Ferrari, el Ing. Civil Diego Pérez y la Arq. Camila Díaz, quienes participaron activamente en las instancias de validación y se tomaron el tiempo para ayudarnos, pese a sus complicadas agendas.

También deseamos expresar nuestro agradecimiento al CIE, en especial a Martín Regueiro y Ximena Scasso, quienes nos brindaron su orientación y apoyo en el desarrollo del emprendimiento y aún continúan haciéndolo.

Además, queremos mencionar a los revisores, Martín Solari, Leonardo Scafarelli y Rafael Bentancur, quienes nos brindaron su guía y consejos.

Finalmente, agradecer a todos los docentes, ayudantes y funcionarios de la Universidad ORT que han sido parte de nuestro proceso de formación académica y profesional durante la carrera.

Nuevamente, a todas estas personas, ¡muchas gracias por acompañarnos en este proceso y por sus invaluable aportes!

Abstract

El rubro de la construcción es uno de los que más rápido se expande en Uruguay. Hoy en día, este sector emplea a más de 50.000 personas (únicamente considerando aquellas en relación de dependencia) y es uno de los que más demanda de trabajo tiene en comparación a cualquier otra industria, ya que cada vez se construye más, por lo que cada vez se necesita mayor cantidad de gente capacitada en el rubro. [1]

Cuando hablamos de capacitación, hablamos de educación, la cual es de suma importancia. Además, el conocimiento es una de las principales fuentes de divisas del país. El problema es que, en este sector, educarse y estar formado es cada vez más difícil; hay muchas personas a las cuales les toma muchos años formalizarse, aunque esto no les asegura un trabajo inmediato. Incluso una gran cantidad tienen la necesidad de salir a trabajar sin haber terminado sus estudios.

Estas personas quieren realizar cursos para poder especializarse o actualizarse en el rubro, pero debido a restricciones con sus horarios laborales, les es muy complicado sobrellevar aquellos con horarios fijos. Además, estos contienen un temario extenso y genérico, lo que resulta en que no los completen enteramente, adquiriendo conocimientos parcialmente e impactando negativamente en su aprendizaje y capacitación, pudiendo traer problemas de seguridad, mayor riesgo de accidentes laborales y menor productividad.

Aquí es donde tiene lugar nuestra propuesta: TechBuildMe. Una plataforma web de aprendizaje virtual, orientada al rubro de la construcción, donde las personas con cierta experiencia, técnicos y profesionales relacionados al mismo que quieren adquirir, profundizar conocimientos o compartir los ya existentes, pueden hacerlo. Esto a través de la realización de cursos cortos, específicos en temas que son aplicables en este sector. Dichos cursos incluyen características como recorridos virtuales en obras, vídeos con distintos ángulos de cámara y un mecanismo que permite a los docentes que brindan cursos el corregir trabajos entregables, así también como otorgar feedback al estudiante.

El resultado final de este proyecto consta de un MVP validado por nuestros interesados, el cual se constituye de una plataforma web con las funcionalidades

mencionadas, basada en Serverless Framework, utilizando los servicios de AWS y con una arquitectura multi-tenant, permitiendo separar la información sensible de cada cliente de manera simple.

Como próximos pasos, el equipo buscará, junto con el Centro de Innovación y Emprendimientos (CIE), validar el producto con clientes del mercado, como pueden ser empresas constructoras.

Palabras clave

API, AWS, Construcción, CDN, Deploy, Emprendimiento, Endpoint, Framework, Git, GitHub, GitHub Actions, Interesados, JavaScript, MVP, Node.js, Plataforma Web, React.js, Recorridos Virtuales, REST API, SCRUM, Serverless Framework, TypeScript, Validación.

Glosario

Android: Sistema operativo móvil desarrollado por Google para diversos dispositivos, famoso por su versatilidad, personalización y amplia variedad de aplicaciones.

ANII: Por sus siglas, Agencia Nacional de Investigación e Innovación de Uruguay, es la encargada de promover y financiar la investigación científica y tecnológica.

API: Por sus siglas, Interfaz de Programación de Aplicaciones, la cual provee un conjunto de reglas y protocolos que permiten a las aplicaciones interactuar entre sí para intercambiar datos y funcionalidades de forma segura y eficiente.

Aplicación web (Webapp): Programa o aplicación que se ejecuta en un navegador web.

AWS: Por sus siglas, Amazon Web Services, es la plataforma de servicios en la nube de Amazon.

Backend: Parte de una aplicación web que se encarga de procesar la lógica del servidor/es y la/s base/s de datos.

Backlog: Lista de tareas pendientes por realizar en un proyecto.

Bug: Error o fallo en el código de una aplicación.

CDN: Por sus siglas, Red de Distribución de Contenidos, es un sistema de servidores distribuidos geográficamente que almacenan y distribuyen contenido web de manera eficiente y escalable, acelerando la entrega de contenidos a los usuarios finales.

CIE: Por sus siglas, Centro de Innovación y Emprendimientos de la Universidad ORT Uruguay, tiene como objetivo promover la innovación y el emprendimiento en la comunidad universitaria.

Concepto de “admin”: Usuario que, además de contar con las funcionalidades de “estudiante” y “docente” (excluyendo solicitar la creación de un curso), tiene la habilidad de crear, modificar cursos y ver un listado de las solicitudes para crear un curso.

Concepto de "docente": Usuario que además de contar con las funcionalidades de un "estudiante", está a cargo de uno o más cursos en nuestra plataforma web, pudiendo responder comentarios y dar feedback de ejercicios a los estudiantes.

Concepto de "estudiante": Usuario que puede consumir los cursos de nuestra plataforma web, entregar ejercicios y tener cursos favoritos. A su vez, puede solicitar la creación de cursos para convertirse en "docente" de uno.

Endpoint: Punto final de una API al que se puede acceder mediante una solicitud HTTP.

Feedback: Respuesta o retroalimentación proporcionada a una persona o grupo en respuesta a una acción o rendimiento.

Framework: Estructura o conjunto de herramientas y librerías utilizadas para desarrollar una aplicación.

Frontend: Parte de una aplicación web que se encarga de la interfaz de usuario y la interacción con el usuario.

Git: Sistema de control de versiones utilizado para rastrear cambios en el código de una aplicación.

GitHub: Plataforma de alojamiento de repositorios de Git en línea.

GitFlow: Método de trabajo para el control de versiones en Git que utiliza ramas separadas para cada característica o corrección de errores.

Google Chrome: Navegador web desarrollado por Google, conocido por su rapidez, estabilidad y gran compatibilidad con diversas plataformas.

iOS: Sistema operativo móvil desarrollado por Apple para sus dispositivos, reconocido por su simplicidad, seguridad y alta calidad en la experiencia de usuario.

JIRA: Herramienta de gestión de proyectos utilizada para rastrear y gestionar tareas en un equipo de desarrollo.

macOS: Sistema operativo de escritorio desarrollado por Apple para sus ordenadores Mac, destacado por su diseño elegante, facilidad de uso y fluidez en el rendimiento.

Mínimo producto viable (MVP): Producto con un conjunto mínimo de características suficientes para satisfacer las necesidades de los primeros usuarios y validar la idea del producto.

OS: Por sus siglas, Sistema operativo, es el software que gestiona los recursos de un ordenador o dispositivo móvil y permite la ejecución de programas y aplicaciones.

Prototipo: Modelo inicial o representación temprana de un producto o idea.

Pull Request (PR): Solicitud para fusionar cambios de una rama en otra en un repositorio de Git.

QA: Control de calidad, proceso de revisión y pruebas de una aplicación.

Rama: Versión separada de un repositorio de Git que contiene cambios específicos.

Redux: Biblioteca de JavaScript utilizada para gestionar el estado de una aplicación.

Release: Versión específica de una aplicación que se ha lanzado públicamente.

Responsive: Diseño de una página web que se adapta a diferentes tamaños de pantalla.

Safari: Navegador web desarrollado por Apple, caracterizado por su eficiencia y seguridad en los dispositivos de la marca.

SPA: Por sus siglas, Aplicación de una Sola Página, utiliza una sola página HTML y actualiza el contenido dinámicamente.

Sprint: Período de tiempo limitado durante el cual se realizan tareas específicas en un proyecto de desarrollo.

Template: Plantilla o modelo utilizado como base para crear nuevos documentos o diseños.

Trade-off: Toma de decisiones entre dos o más opciones que implican beneficios y costos.

URL: Por sus siglas, Localizador Uniforme de Recursos, es la dirección que identifica una página web en la red.

UI: Por sus siglas, Interfaz de Usuario, es el aspecto visual y la interacción que tienen los usuarios con una aplicación.

User Story: Descripción breve y sencilla de una característica o funcionalidad de una aplicación desde la perspectiva del usuario.

Índice

1. Introducción.....	16
1.1. Presentación del equipo.....	16
1.2. Selección del proyecto.....	17
1.3. Objetivos.....	18
1.3.1. Objetivos académicos.....	19
1.3.2. Objetivos del producto.....	19
1.3.3. Objetivos del proyecto.....	20
1.4. Estructura del documento.....	20
2. Problema.....	23
2.1. Contexto.....	23
2.2. Descripción del problema.....	23
2.3. Análisis de interesados.....	24
2.4. Interesados que nos acompañaron.....	26
2.5. Desafíos del proyecto.....	27
3. Solución.....	29
3.1. Solución propuesta.....	29
3.2. Principales funcionalidades.....	30
3.3. Alcance del proyecto.....	32
4. Ideación inicial.....	34
4.1. Relevamiento de datos.....	34
4.1.1. Entrevistas.....	34
4.1.2. Encuestas.....	35
4.1.3. User Persona.....	37
4.1.4. Benchmarking.....	39
4.2. Análisis del mercado.....	40
4.3. Público objetivo.....	42
4.4. Modelo de negocio Canvas.....	43
4.5. Nombre y logo.....	45
5. Marco metodológico.....	47
5.1. Características del proyecto.....	47
5.2. Características del equipo.....	48
5.3. Metodologías de trabajo.....	49

5.3.1. Lean Startup y Scrum	49
5.3.2. Aplicación de las metodologías.....	50
5.3.3. Ciclo de vida	53
5.4. Roles	54
5.5. Conclusiones de las metodologías usadas	54
6. Ingeniería de requerimientos.....	55
6.1. Proceso de definición de requerimientos	55
6.1.1. Elicitación.....	55
6.1.2. Priorización y reescritura.....	56
6.1.3. Especificación	59
6.2. Requerimientos funcionales	61
6.3. Requerimientos no funcionales	62
6.4. Definición de MVP	64
6.5. Conclusiones.....	65
7. Arquitectura y diseño.....	66
7.1. Descripción general de la Arquitectura.....	66
7.1.1. Consideraciones arquitectónicas	66
7.1.2. Visión general	66
7.2. Atributos de calidad.....	69
7.2.1. Deployabilidad.....	69
7.2.2. Disponibilidad.....	70
7.2.3. Eficiencia.....	71
7.2.4. Modificabilidad	73
7.2.5. Seguridad.....	75
7.3. Estándares	76
7.3.1. Sistema de diseño.....	76
7.3.2. Manejo de datos en el Frontend.....	78
7.3.3. REST API.....	78
7.3.4. Cumplimiento de Twelve Factor App	79
7.3.5. Arquitectura Multi-Tenant.....	82
7.4. Tecnologías.....	83
7.5. Gestión de datos	84
7.6. Detalles de la arquitectura.....	86

7.6.1. Vista de módulos.....	86
7.6.2. Flujo de publicación de un nuevo curso	91
7.7. Conclusiones.....	92
8. Gestión de proyecto	93
8.1. Planificación del proyecto.....	93
8.2. Herramientas utilizadas.....	96
8.3. Metodologías de trabajo.....	97
8.3.1. Seguimiento del trabajo	97
8.3.2. Planificación y seguimiento de la etapa de construcción	99
8.3.3. Estimación de las tareas	101
8.4. Plan de Releases	102
8.5. Principales hitos del proyecto.....	104
8.6. Métricas.....	106
8.6.1. Gestión.....	106
8.6.2. Construcción	108
8.7. Gestión de la comunicación	111
8.8. Gestión de riesgos	111
8.8.1. Detección	111
8.8.2. Estrategia.....	112
8.8.3. Seguimiento	118
8.9. Conclusiones.....	121
9. Gestión de la configuración	122
9.1. Herramientas utilizadas.....	122
9.1.1. Código.....	122
9.1.2. Documentación	123
9.2. Organización de repositorios.....	125
9.3. Control de versiones	127
9.3.1. Código.....	127
9.3.2. Documentación	130
9.4. Conclusiones.....	130
10. Gestión de calidad.....	131
10.1. Definición de calidad para TechBuildMe	131
10.2. Prácticas de aseguramiento de la calidad.....	132

10.2.1. Uso de estándares	132
10.2.2. Revisiones	134
10.2.3. Tests unitarios.....	135
10.3. Validaciones	136
10.3.1. Prototipos.....	137
10.3.2. Requerimientos funcionales.....	138
10.3.3. Releases	138
10.3.4. MVP	141
10.4. Cumplimiento de atributos de calidad	143
10.5. Bugs	144
10.6. Conclusiones.....	146
11. Conclusiones finales.....	147
11.1. Conclusiones generales	147
11.2. Lecciones aprendidas	149
11.3. Próximos pasos.....	149
12. Referencias bibliográficas	151
13. Anexos.....	154
13.1. Anexo 1: Entrevistas realizadas	154
13.2. Anexo 2: Encuesta realizada.....	155
13.3. Anexo 3: Documentación CIE	162
13.4. Anexo 4: Prototipos realizados.....	162
13.5. Anexo 5: RF iniciales.....	163
13.6. Anexo 6: Desglose de US	163
13.7. Anexo 7: Catálogo de elementos del diagrama general.....	185
13.8. Anexo 8: Vistas de componentes y conectores	187
13.9. Anexo 9: Estructura de datos en Redux.....	189
13.10. Anexo 10: Endpoints	191
13.11. Anexo 11: Tablas y Colecciones	194
13.12. Anexo 12: Crear un nuevo producto.....	201
13.13. Anexo 13: Diagrama de paquete de capas	202
13.14. Anexo 14: Versiones de la planificación anual	202
13.15. Anexo 15: Detalle de Sprints	205
13.16. Anexo 16: Registro detallado de horas de trabajo.....	216

13.17. Anexo 17: Evolución de los riesgos.....	216
13.18. Anexo 18: Template de Pull Request.....	227
13.19. Anexo 19: Librerías externas.....	228
13.20. Anexo 20: Feedback de Releases.....	230
13.21. Anexo 21: Tareas de las validaciones.....	230
13.22. Anexo 22: Cumplimiento de las Heurísticas de Nielsen (RNF5)	232
13.23. Anexo 23: Cumplimiento de responsiveness (RNF6).....	239
13.24. Anexo 24: Cumplimiento de compatibilidad con navegadores (RNF8)	241

1. Introducción

El presente documento describe el proyecto “TechBuildMe: plataforma de aprendizaje para el rubro de la construcción”, realizado como requisito para la obtención del título Ingeniero de Software de la Universidad ORT Uruguay.

TechBuildMe (TBM) es una plataforma web de aprendizaje orientada al sector de la construcción, para que las personas con cierta experiencia, técnicos y profesionales relacionados al mismo, que quieran adquirir, profundizar conocimientos o compartir los ya existentes, pueden hacerlo.

El objetivo de nuestro MVP, resultado de este proyecto, es mejorar los procesos educativos y la inserción laboral en las personas que quieren ingresar en el rubro de la construcción o las que ya se encuentran en éste y quieren especializarse en cierta área. A su vez, facilitar el acceso y aprendizaje del conocimiento básico y clave para que la inserción laboral sea mucho más rápida, siendo el mismo impartido por expertos del dominio. De esta forma, ayuda a que la educación online sea una adaptación asociada a temas relevantes que brinden una mejor efectividad.

En este capítulo, detallaremos los integrantes del equipo, cómo surgió la idea para este proyecto, nuestro público objetivo, los objetivos planteados y la estructura que tendrá el documento en los siguientes capítulos.

1.1. Presentación del equipo

A continuación, presentamos a los integrantes que conforman el equipo del proyecto.



Figura 1 - Martina Severo



Figura 2 - Santiago Tonarelli

- Martina Severo (Figura 1): Licenciada en Ingeniería de Software y estudiante de la carrera de Ingeniería en Sistemas de la Universidad ORT. Tiene más de 1 año y medio de experiencia como Software developer. Actualmente trabaja como Software developer en Qualabs. En adición, forma parte de la cátedra de Teoría de la Computación de la Universidad ORT.
- Santiago Tonarelli (Figura 2): Licenciado en Ingeniería de Software y estudiante de la carrera de Ingeniería en Sistemas de la Universidad ORT. Tiene más de 2 años de experiencia como Software developer. Actualmente trabaja en CódigoDelSur y cuenta con una certificación en docencia expedida por la Universidad ORT. En adición, forma parte de la cátedra de Teoría de la Computación y de la cátedra de Ingeniería de Software de la Universidad ORT.

El equipo ha trabajado en conjunto desde los primeros semestres de la carrera, lo que resulta en algo ventajoso a la hora de organizar, planificar y llevar adelante las tareas, ya que hay buena comunicación y confianza entre ambos, pudiendo complementarse de manera favorable.

1.2. Selección del proyecto

Desde el inicio de nuestra carrera, ambos estuvimos al tanto del apoyo que brinda la Universidad ORT con relación a proyectos finales de carrera con enfoque emprendedor, siendo el Centro de Innovación y Emprendimientos (CIE) uno de estos mecanismos de soporte. Esta oportunidad coincidió con nuestro interés en utilizar las herramientas de software y conocimientos adquiridos durante nuestra formación académica para emprender.

A partir de junio de 2021, nuestro equipo comenzó a realizar análisis exhaustivos sobre diversos problemas que se manifestaban como necesidades para ciertas personas. Finalmente, estos análisis convergieron en la idea central de nuestro proyecto actual.

En adición, uno de los integrantes tiene a alguien cercano relacionado al sector de la construcción, quien a su vez nos ayudó a conectar con otras personas también en este rubro. De esta forma pudimos validar con dichos individuos, mediante entrevistas,

el problema que existe a la hora de actualizar o adquirir nuevos conocimientos relativos al sector de la construcción.

A su vez, teniendo en cuenta que la educación es de suma importancia, ya que abre puertas y genera empleos, consideramos que nuestro proyecto ayuda a mejorar los procesos educativos y la inserción laboral. Esto enfocado en las personas que quieren ingresar en el rubro de la construcción o las que ya se encuentran en éste y quieren especializarse en cierta área. Así, facilitamos el acceso y aprendizaje del conocimiento básico y clave para que su inserción laboral sea mucho más rápida, de modo que el conocimiento sea impartido por expertos del dominio, contribuyendo a que la educación online sea una adaptación asociada a temas relevantes que brinden una mejor efectividad.

Además, el hecho de que el proyecto esté orientado hacia el rubro de la construcción, que es uno de los más grandes mundialmente e involucra a muchas personas, nos permitirá ayudar a mejorar el crecimiento profesional de dichos individuos, proveyéndoles un servicio que cubra sus necesidades. Asimismo, desde que comenzó la pandemia, los cursos virtuales se vieron potenciados y las personas se han ido acostumbrando a ellos, pero también tienen grandes carencias y aspectos sin cubrir, los cuales buscamos subsanar con este proyecto.

Por otro lado, a nivel académico, el poder crear un proyecto emprendedor nos otorga mucha experiencia y nos permite organizarnos mejor, a la vez que ponemos en práctica los conocimientos adquiridos a lo largo de la carrera, pudiendo ser responsables de varios aspectos, no solo del desarrollo tecnológico de dicho proyecto.

1.3. Objetivos

Debido a la naturaleza emprendedora del proyecto, al principio establecimos objetivos que se escapaban del alcance y eran de alto nivel. Fue por ello que, luego de un período de refinamiento en las etapas iniciales del proyecto, los ajustamos, obteniendo los mencionados a continuación.

1.3.1. Objetivos académicos

- **Aplicar prácticas de Ingeniería de Software aprendidas durante la carrera**

Debido a que se trata de un proyecto final de carrera, consideramos que aplicar lo aprendido a lo largo de la carrera en cuanto a prácticas de ingeniería de software es una buena oportunidad para profundizar en su aplicación y aprovechar lo que nos otorgó la Universidad ORT.

- **Poner en práctica una metodología ágil vista en la carrera**

A lo largo de la carrera, hemos estudiado distintas metodologías de desarrollo, entre ellas las ágiles. A su vez, en nuestro aspecto laboral, hemos podido experimentarlas desde la perspectiva del desarrollador. Por estas razones, nos parece clave, como cierre a esta etapa académica, poner en práctica una de estas metodologías, desde una perspectiva general.

1.3.2. Objetivos del producto

- **Validar que la idea aporte valor a nuestro público objetivo a través de la creación de un MVP**

Determinar si nuestra propuesta de valor es relevante para nuestro público objetivo. Al desarrollar un MVP, podemos identificar de manera temprana si nuestra idea resuelve un problema específico, satisface una necesidad real o genera un interés genuino en nuestro público objetivo.

El proceso de validación de la idea es esencial para garantizar que nuestros recursos y esfuerzos se estén enfocando en una solución viable y que agregue valor al mercado. Al llevar a cabo esta validación, también podemos obtener información valiosa sobre cómo mejorar nuestro producto y adaptarlo a las necesidades de nuestros clientes potenciales. Consiguiendo así con esta validación maximizar la probabilidad de éxito en el futuro lanzamiento de nuestro producto al mercado.

1.3.3. Objetivos del proyecto

- **Construcción de una plataforma web de calidad**

Desarrollar una solución que cumpla con los estándares de calidad establecidos en la industria. La calidad de la plataforma web está relacionada con su capacidad para satisfacer las necesidades y expectativas de los usuarios.

Para lograrlo, se deben considerar atributos de calidad como la Usabilidad, Eficiencia, Seguridad, Disponibilidad, Portabilidad y Modificabilidad [2], así como estándares adoptados por la industria para plataformas web con estas características. Además, tener en cuenta el uso de tecnologías y herramientas de desarrollo adecuadas, la adopción de las mejores prácticas y metodologías de desarrollo y pruebas, y el llevar a cabo procesos de control de calidad.

1.4. Estructura del documento

A continuación, brindamos detalle acerca de cómo está estructurado todo el documento, explicando brevemente cada sección contenida.

- **Introducción**

Contiene una presentación de los integrantes, cómo surgió la idea para el proyecto, los objetivos definidos (académicos, del producto y del proyecto) y como está estructurado el documento.

- **Problema**

En esta sección describimos el problema identificado, incluyendo el contexto y la descripción del mismo, los interesados con quienes validamos y que desafíos presenta el proyecto.

- **Solución**

Aquí describimos la solución creada para enfrentar el problema, sus principales funcionalidades y el alcance del proyecto.

- **Emprendimiento**

Engloba la descripción de los procesos que llevamos a cabo para relevar los datos, definir nuestro público objetivo, realizar un análisis del mercado y un modelo de negocio Canvas. También contiene el nombre y logo que creamos para la solución.

- **Marco metodológico**

Abarca explicaciones acerca de las características del proyecto y del equipo, que metodologías de trabajo utilizamos, como organizamos los roles y que conclusiones obtuvimos al poner en práctica dichas metodologías.

- **Ingeniería de requerimientos**

Contempla detalles sobre los mecanismos para la obtención de los requerimientos funcionales y no funcionales, así como la definición de nuestro MVP.

- **Arquitectura y diseño**

Contiene descripciones relativas a las transformaciones que fue teniendo la arquitectura y diseño de nuestra solución, incluyendo diagramas para mejor entendimiento.

- **Gestión de proyecto**

Esta sección explica como gestionamos nuestro proyecto a lo largo de las diferentes etapas que realizamos durante su proceso, junto con la evidencia correspondiente.

- **Gestión de la configuración**

Menciona las herramientas que utilizamos para el manejo y control de versiones, incluyendo lo que esto implica.

- **Gestión de la calidad**

Esta sección está orientada a detallar cual es nuestra definición de calidad y las actividades que llevamos a cabo para asegurar la calidad de nuestro producto, así como la importancia de la misma y los bugs encontrados.

- **Conclusiones**

Aquí comentamos los resultados que obtuvimos al realizar este proyecto, contemplando distintos aspectos mencionados en secciones anteriores. Además, planteamos nuestras conclusiones generales, que lecciones aprendimos en base a lo realizado y cuales son nuestros próximos pasos.

2. Problema

Este capítulo tiene como finalidad introducir al lector al problema, comentando en primer lugar, su origen y las bases para fundamentar la existencia del mismo. Luego pasaremos a describir el problema en sí y quienes se ven afectados.

Posteriormente, mencionaremos los principales interesados que identificamos. Pudimos acercarnos a un grupo que nos fue de ayuda para la etapa de definición del problema y algunos continuaron brindando su apoyo para validar la solución.

Finalmente, comentaremos acerca de los desafíos que se nos presentaron a la hora de realizar el proyecto, siendo el principal, la falta de conocimiento del rubro por parte de los integrantes.

2.1. Contexto

Hoy en día, el sector de la construcción es uno de los que más relevancia tiene en lo referente a la economía de Uruguay. Según un artículo publicado por el CEEIC (Centro de Estudios Económicos de la Industria de la Construcción): "...tiene un rol clave en la producción de bienes de inversión como la infraestructura de transporte, el saneamiento y agua potable, montaje de plantas industriales o la producción y reparación de viviendas...". Además, menciona que cuanto más eficiencia y mejor funcionamiento tiene el sector, mayor impacto positivo genera en el bienestar de los hogares [3].

Otro punto para destacar es que el rubro de la construcción es uno de los que más rápido se expande en Uruguay, empleando a más de 100.000 personas a lo largo de los años. A su vez, es uno de los que más demanda de trabajo tiene en comparación a cualquier otra industria, ya que la cantidad de construcciones van en aumento y por lo tanto, cada vez se necesita de más gente capacitada en el rubro.

2.2. Descripción del problema

Cuando mencionamos que el sector de la construcción requiere de personas capacitadas, también nos referimos a la educación de dichos individuos, la cual tiene un rol significativo. El problema es que, en este rubro, educarse y estar formado es

cada vez más complejo, ya que hay mucha gente a la cual les toma bastantes años formalizarse, y aun así, esto no les asegura un trabajo inmediato. Incluso una gran cantidad tienen la necesidad de salir a trabajar sin haber terminado sus estudios.

Estas personas quieren realizar cursos para poder especializarse o actualizarse en el rubro, pero debido a restricciones con sus horarios laborales, les es muy complicado sobrellevar cursos con horarios fijos y/o que requieran presencialidad. Además, estos contienen un temario extenso y genérico, lo que resulta en que no los completen en su totalidad, adquiriendo conocimientos parcialmente e impactando negativamente en su aprendizaje y capacitación.

Entonces, debido a que para poder trabajar en el sector de la construcción se requieren un conjunto de habilidades técnicas importantes, si las personas en este sector no están capacitadas correctamente, esto puede traer problemas de seguridad, mayor riesgo de accidentes laborales y menor productividad.

2.3. Análisis de interesados

En esta sección detallamos a nuestros interesados mediante el uso de una matriz de poder/interés (Tabla 1), agrupándolos según su nivel de autoridad o “poder” y su nivel de preocupación o “interés” respecto al proyecto. Dependiendo de donde se sitúen en la matriz, corresponde aplicar cierta estrategia para mantenerlos involucrados.

PODER	Alto	Involucrar, mantener satisfechas	Involucrar y atraer activamente
	Bajo	Monitorear	Mantener informadas
		Bajo	Alto
INTERÉS			

Tabla 1 - Matriz de poder / interés

A continuación, los grupos de interesados identificados (Tabla 2):

Interesados	Poder	Interés	Estrategia
Empresas constructoras	Alto	Bajo	Mantener satisfechas
Instituciones relacionadas y que apoyan al rubro de la construcción	Alto	Bajo	Mantener satisfechas
Estudiantes y trabajadores del rubro (NO creadores de contenido)	Bajo	Alto	Mantener informados
Estudiantes y trabajadores del rubro (creadores de contenido)	Alto	Alto	Atraer activamente
Inversores	Alto	Alto	Atraer activamente
Patrocinadores	Alto	Bajo	Mantener satisfechos

Tabla 2 - Interesados identificados

- **Empresas constructoras**

Las empresas constructoras tienen experiencia y renombre en el rubro y gestionan todo tipo de obras y personal relacionado a la construcción, por lo que su apoyo en el proyecto ayudará a brindar credibilidad y confianza a nuestro producto. Por este motivo, tienen un poder alto, ya que poseen autoridad sobre el proyecto para que se adecue a sus necesidades. Su interés es bajo porque su preocupación por las decisiones internas del equipo es escasa.

- **Instituciones relacionadas y que apoyan al rubro de la construcción**

Al igual que el caso anterior, estos entes serán de ayuda para crear convenios que respalden nuestro producto, otorgando a nuestro contenido una mayor credibilidad. Por ello, su grado de poder es alto, ya que pueden tener influencia sobre la toma de ciertas decisiones, mientras que su interés es bajo.

- **Estudiantes y trabajadores del rubro (no creadores de contenido)**

Este grupo comprende a aquellas personas que quieren adquirir conocimientos y consumir contenido de nuestro producto, pero no están interesadas o no tienen la experiencia suficiente para crearlo. Su poder es bajo, ya que no tienen influencia sobre

el producto, pero su interés es alto debido a que se ven afectados directamente por las decisiones relativas al mismo.

- **Estudiantes y trabajadores del rubro (creadores de contenido)**

A diferencia del conjunto anterior, estos individuos pueden querer especializarse, pero a la vez también desean brindar contenido por medio de nuestro producto. De esta manera tendrían un “rol docente” y podrían, a modo de ejemplo, crear cursos. Por lo tanto, su poder es alto, ya que son los que otorgan contenido a nuestro producto. Su interés también es alto; tienen influencia en las decisiones, de forma que estén involucrados y se mantengan interesados en el producto.

- **Inversores**

Son aquellas personas que proveen ayuda económica al proyecto, invirtiendo su propio dinero en busca de algún tipo de beneficio o rentabilidad a futuro. Por este motivo, su poder e interés son altos, ya que tienen autoridad sobre las decisiones y a su vez, están pendientes de las mismas porque se ven afectados directamente.

- **Patrocinadores**

Comprende al grupo de individuos que ayudan a expandir la voz sobre nuestro producto, como puede ser a través de publicidades o propagandas, de modo que atraigamos más público. Su poder es alto debido a que proveen financiación del producto y tienen influencia en las decisiones, mientras que su interés o nivel de preocupación es bajo.

2.4. Interesados que nos acompañaron

Para las instancias de validación, conseguimos el apoyo inicial de cuatro personas de las que entrevistamos, las cuales accedieron a colaborar y darnos sus opiniones acerca de nuestro producto a lo largo de todo el proyecto (Tabla 3). Sin embargo, en cierto punto del mismo, una de ellas no pudo continuar brindándonos su apoyo debido a su trabajo y otros compromisos, por lo cual, contamos con el acompañamiento de las tres restantes personas a lo largo del proyecto.

Interesados	Edad	Profesión y cargo	Participación
Fernanda Ferrari	53	Arquitecta, jefa de obra en SABYL S.A.	Todas las validaciones.
Diego Pérez	28	Ingeniero civil, ingeniero en dirección de obra en Administración Nacional de Puerto.	Todas las validaciones.
Camila Díaz	28	Arquitecta, jefa de obra en Faber Proyectos S.R.L.	Todas las validaciones.
Andrés Pérez	26	Ingeniero civil, gerente de obra en RDA Project Management.	Validación de prototipos y RF.

Tabla 3 - Interesados que nos acompañaron en el proyecto

Sumado a estos interesados, nos acompañó Laura Bentos, la cual forma parte de RRHH de la empresa constructora Stiler. Sin embargo, durante la etapa de construcción, se perdió el contacto, debido a que el equipo priorizó el desarrollo y avance del proyecto académico frente al emprendimiento.

Como nota aparte, a este grupo de individuos los referenciamos como nuestros interesados a lo largo del documento.

2.5. Desafíos del proyecto

- **Cumplimiento de objetivos definidos**

El cumplimiento de los objetivos se planteó como uno de los principales desafíos, ya que, al tratarse de un proyecto emprendedor, se requirió ajustarse a los requerimientos y preferencias de nuestros interesados, cumpliendo a su vez con el alcance y relevancia que requiere un proyecto final de carrera.

- **Contar con interesados**

Dado que nuestro proyecto requiere de validaciones constantes para asegurar que se cumplan las necesidades de nuestro público objetivo, asegurar contar con personas del sector que pudieran garantizarnos su apoyo y disponibilidad a lo largo del proyecto representó un gran reto.

- **Contenido ofrecido por terceros**

El hecho de que la plataforma brinde cursos virtuales dictados por terceros, profesionales en el rubro de la construcción, es un desafío ya que implica lograr apoyo de personas que estén aptas para brindar cursos.

- **Poco conocimiento del tema**

La creación y contenido del producto representó un desafío, ya que debíamos utilizar información lo más realista posible para obtener mejores resultados en las validaciones. No obstante, la realización de entrevistas y encuestas resulta una herramienta de gran ayuda para recolectar más datos acerca de nuestro público objetivo y de temas relacionados al rubro.

3. Solución

En este capítulo, detallaremos la solución inicial y luego la solución que decidimos implementar, mencionando las razones que nos llevaron a tomar ciertas decisiones para modificar el producto.

A su vez, basándonos en la solución implementada, explicaremos las principales funcionalidades, las cuales forman parte de nuestro MVP y posteriormente, describiremos el alcance de nuestro proyecto.

3.1. Solución propuesta

Nuestra solución, nombrada TechBuildMe (TBM), consiste en una plataforma web de aprendizaje virtual, orientada al sector de la construcción. En ella, las personas con cierta experiencia, técnicos y profesionales relacionados a este rubro, que quieren adquirir, profundizar conocimientos o compartir los ya existentes, pueden hacerlo.

La plataforma se basa en brindar contenido a través de la realización de cursos cortos, específicos en temas que son aplicables en este sector, de modo de evitar que se vuelvan tediosos y puedan completarse totalmente con facilidad. De esta forma, aquellos individuos que poseen experiencia y quieren ofrecer sus aprendizajes, pueden crear cursos para que otros los vean y puedan seguir profundizando sus conocimientos.

Los cursos incluyen características como recorridos virtuales en obras, vídeos con distintos ángulos de cámara y un mecanismo que permite que los docentes que brindan los cursos puedan delegar a otros profesionales del rubro, expertos en la temática, la corrección de trabajos entregables, así también como otorgar un feedback al estudiante.

En definitiva, nuestra solución tiene tres roles principales. Por un lado, tenemos al estudiante, que es quien consume el contenido ofrecido y tiene la posibilidad de realizar ejercicios y recibir feedback. Por otro lado está el docente, quien puede solicitar la creación de contenido (como son los cursos) pasando a ser el encargado del mismo, una vez publicado en la plataforma. A su vez, puede contestar preguntas sobre sus cursos y corregir ejercicios hechos por estudiantes para otorgarles

comentarios y ayudar en su aprendizaje. Por último, se encuentra al administrador. Este rol comparte las características de los dos roles mencionados anteriormente, adicionando la creación, modificación y publicación de cursos en el sistema. Es quien recibe las solicitudes de los docentes, las evalúa y se pone en contacto con ellos para llevar a cabo la creación del curso.

Algo importante a destacar es que un usuario puede tomar los tres roles a la vez.

3.2. Principales funcionalidades

A continuación, detallamos las funcionalidades más importantes que componen nuestra solución:

- **Visualización de cursos**

Como mencionamos anteriormente, los estudiantes podrán realizar cursos cortos, con temarios específicos del rubro de la construcción. Estos cursos están conformados por módulos, que a su vez contienen lecciones. A modo de ejemplo, un curso puede desestructurarse en dos módulos, el primero conteniendo siete lecciones y el segundo seis.

Cada lección puede ser de un tipo de cuatro existentes: video, recorrido virtual, ejercicio entregable o material informativo. Para las de tipo video, si el mismo lo permite, se puede ver en varios ángulos de cámara, pudiendo optar por mostrar uno, dos o tres ángulos a la vez. Además, estos ángulos pueden ser intercambiados para verse en el reproductor principal o en uno de los secundarios. Esto en caso de mostrar tres ángulos, de lo contrario, serán dos reproductores iguales o uno.

En el caso de ser de tipo recorrido virtual de una obra de construcción, solo se muestra un video, en el cual se permite girar la cámara 360 grados en cualquier dirección. A su vez, si están disponibles, a lo largo del recorrido virtual, se mostrarán diálogos emergentes con preguntas relacionadas al video, de modo de hacer la experiencia más dinámica y poner en práctica los conocimientos. Las preguntas son del tipo múltiple opción o verdadero o falso, incluyendo también una breve explicación de la respuesta correcta una vez que se responda y pueden deshabilitarse en caso de querer ver el recorrido virtual de manera continua.

Para las lecciones de tipo ejercicio entregable, los estudiantes pueden descargar la letra de la consigna y subir un documento adjunto brindando su solución, para posteriormente poder recibir feedback por parte de un docente. Por último, para las de tipo material informativo, se puede descargar el documento conteniendo información de utilidad referida a la lección.

- **Recorridos virtuales en obras**

Como ya mencionamos anteriormente, los recorridos virtuales son uno tipo de los que puede ser una lección, los cuales permiten girar el ángulo de la cámara 360 grados. Esta es la funcionalidad de mayor valor de nuestro proyecto, ya que involucra hacer que nuestro producto sea compatible con videos 360, permitiendo al usuario observar libremente lo que ocurre en cualquier dirección en la obra, como si estuviera físicamente en el lugar. Asimismo, puede visualizar preguntas relacionadas al curso durante el recorrido para practicar lo aprendido y hacer la experiencia más interactiva, a la vez que se asemeja lo más posible a la realidad.

Un comentario importante es que si el dispositivo en que se esté visualizando el recorrido virtual cuenta con un giroscopio y el navegador es compatible con esta herramienta, se podrá realizar el movimiento de la cámara con el desplazamiento del dispositivo en cuestión. A modo de ejemplo, si se utiliza desde un móvil, no será necesario movilizar la cámara manualmente, sino moviendo y girando el dispositivo. A su vez, este puede acoplarse a algún tipo de soporte compatible, como puede ser el casco de realidad virtual Gear VR de Samsung.

- **Ejercicios entregables**

Esta funcionalidad también es uno de los tipos existentes de lección y permite a los estudiantes realizar ejercicios relacionados a un curso, de modo que puedan practicar lo visto en el mismo. Posteriormente a su entrega, podrán revisar el estado de la misma, recibiendo comentarios por parte de un docente, pudiendo así verificar si lo que hicieron es correcto o si hay aspectos que deben revisar. Esto funciona como un seguimiento de su progreso y aprendizajes, optimizando la adquisición de conocimientos a través de nuestro producto.

- **Feedback**

Como mencionamos en el punto anterior, el feedback es otorgado de un docente a un estudiante, mediante la corrección de ejercicios entregables de un curso, siendo el sistema el encargado de notificarle la existencia del mismo. Esto proporciona a los estudiantes una forma de corroborar si su comprensión de los temas vistos es correcta, y si no, tener la chance de pulir esos aspectos.

3.3. Alcance del proyecto

El alcance de nuestro proyecto comprende la realización de un MVP, el cual consta de una plataforma web con las siguientes funcionalidades:

- Listado de cursos disponibles, pudiendo modificarlos en caso de ser administrador.
- Vista previa de un curso, enseñando detalles del mismo, como duración, docente, puntuación, descripción, contenido, opiniones, entre otros. Pudiendo modificarlo o publicarlo en caso de ser administrador.
- Contenido específico de una lección de un curso, contando con reproductores de video, opciones para modificar la disposición de los ángulos, guía de uso, preguntas para recorridos virtuales y documentos. La visibilidad de estos elementos varía según el tipo de lección. También se mostrará información relativa a la descripción, contenido y opiniones del curso al que pertenece la lección.
- Listado de ejercicios entregables que pueden realizarse, en el caso de estudiantes. En el caso de docentes, ejercicios que pueden ser corregidos.
- Detalle de un ejercicio entregable, pudiendo descargar la letra de la consigna, ver comentarios y otros detalles. En el caso de docentes, pueden también agregar los comentarios y subir archivos adjuntos.
- Creación y modificación de un curso, en caso de ser administrador. Pudiendo crear/modificar su estructura de forma completa, guardando el progreso o publicándolo para los demás usuarios.

- Solicitud de creación de curso en caso de no ser administrador.
- Listado de solicitudes para crear un curso, en caso de ser administrador.
- Listado de cursos favoritos.
- Listado de notificaciones de eventos del sistema.
- Registro de un usuario.
- Inicio y cierre de sesión.
- Cambio de contraseña de cuenta en caso de que se olvide la misma.

4. Ideación inicial

La finalidad de este capítulo es agrupar las actividades que realizamos al inicio del proyecto debido a las características emprendedoras del mismo, que sentaron las bases para poder definir nuestra idea.

Aquí detallaremos las diversas técnicas que utilizamos para obtener datos de distintos trabajadores del rubro de la construcción, las cuales nos ayudaron a generar insights.

Dentro de dichas técnicas, destacamos entrevistas y encuestas, la realización de un User Persona para luego poder definir nuestro público objetivo, el estudio del mercado (incluyendo benchmarking) y la creación de un modelo de negocio Canvas (incluyendo la propuesta de valor, basada en los insights que serán detallados en el capítulo [6. Ingeniería de requerimientos](#)).

También mencionaremos brevemente el nombre y logo que diseñamos para nuestro producto.

4.1. Relevamiento de datos

Durante la etapa inicial de nuestro proyecto, recolectamos información acerca del rubro de la construcción a través de la aplicación de diversas técnicas que mencionaremos a continuación. Estos datos luego nos fueron útiles para pulir nuestra idea inicial y así terminar de definirla para seguir con las siguientes etapas del proyecto.

A continuación, explicaremos cada una de dichas técnicas y los resultados obtenidos.

4.1.1. Entrevistas

Hicimos un total de 25 entrevistas a trabajadores del rubro (16 ingeniero/as civiles, 7 arquitecto/as y 2 técnicos constructores), los cuales contaban con distintos niveles de experiencia, algunos habiéndose graduados recientemente y otros formando parte del sector hace varios años. Esto nos permitió recolectar mucha información útil que nos permitió encontrar los puntos débiles y las necesidades del sector. A su vez, debido a la variedad de respuestas, pudimos analizar diferentes puntos de vista y opiniones que nos ayudaron a entender mejor ciertos aspectos del mismo.

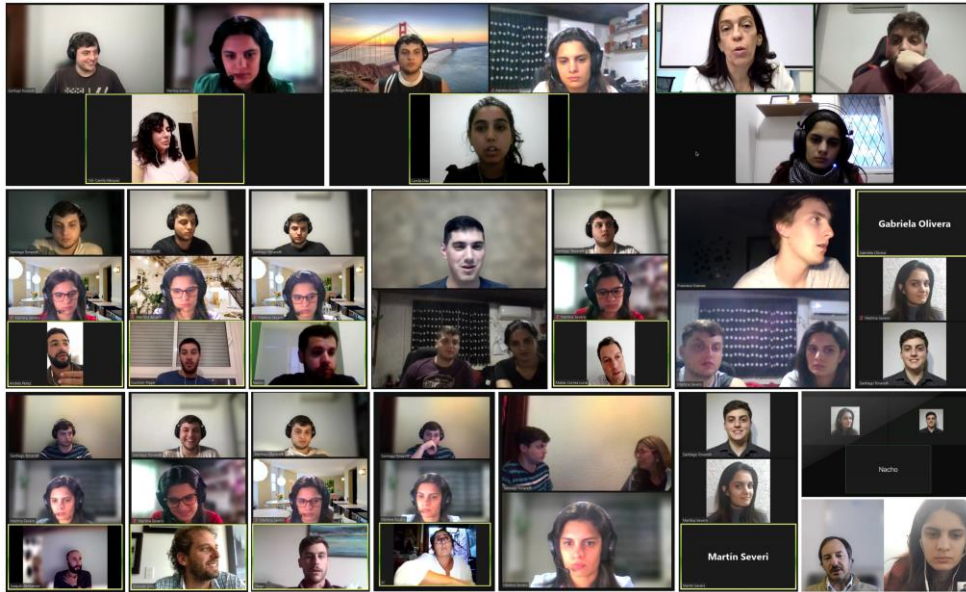


Figura 3 – Algunas de las entrevistas con trabajadores del rubro de la construcción

Las entrevistas y respectivas notas se pueden ver en [13.1. Anexo 1: Entrevistas realizadas.](#)

4.1.2. Encuestas

En conjunto con las entrevistas, realizamos una encuesta, constando de 24 preguntas, enfocadas a conocer sobre experiencias relacionadas a cursos virtuales y sus opiniones sobre ellos. Conseguimos un total de 128 respuestas, donde 63 son ingenieros civiles, 56 arquitectos y 9 técnicos constructores, contando con experiencia desde 1 año hasta 35 años en el rubro.



Figura 4 - Resultados relevantes de la encuesta

A modo de comentario, los resultados de esta encuesta pueden no ser los actuales debido a que la misma continúa abierta y pueden llegar más respuestas. Algunos de los datos más relevantes que obtuvimos, tanto de las entrevistas como de las encuestas, fueron:

- El 85% realizó cursos virtuales para capacitarse o actualizarse en temas relacionados al rubro de la construcción.
- El 84% conocía poco o nada acerca del docente que dictaba el curso.
- Un 59% dijo que los cursos virtuales que realizó no contaban con reseñas (calificaciones y/o comentarios) sobre el dictado del docente.
- En cuanto a que valoraban más de un curso virtual, las respuestas más populares y relevantes, superando el 68%, fueron las de poder verlos cuando quieran y donde sea, y que fueran dados por personas de confianza, expertos en el área.
- Dentro de las frustraciones, un 43% destacó la dificultad de contactar y obtener feedback del docente del curso.
- Un 72% realizó cursos virtuales pagos.
- En promedio, cuentan con aproximadamente 3 horas libres por día.
- Un 55% indicó que estarían dispuestos a brindar cursos como docentes, explicando temas o procesos relacionados al rubro.
- La cantidad de temas mencionados sobre los que podrían enseñar fueron 49.
- Un 52% de los encuestados vieron de gran utilidad el poder obtener vistas desde distintos enfoques en cursos virtuales enfocados a lo práctico.
- 23 de los 25 entrevistados se mostraron disconformes con el hecho de concurrir a obras en el último año de la carrera, teniendo en cuenta que 18 de ellos ya trabajaban en el rubro hace al menos 1 año.
- 21 de los 25 entrevistados manifestaron que la profundidad acerca de lo que aprendieron sobre gestión en el rubro fue muy insuficiente al momento de insertarse en el ámbito laboral. 15 de ellos tuvieron que realizar cursos por fuera para profundizar estos conocimientos.

- 14 de 25 entrevistados debieron abandonar sus cursos o carreras debido a que sus horarios laborales sufrían variaciones que no les permitían continuar con ello.

La encuesta completa con los resultados se puede visualizar en [13.2. Anexo 2: Encuesta realizada.](#)

4.1.3. User Persona

Luego creamos un modelo de User Persona, representando aspectos y características relevantes en base a lo que obtuvimos de las entrevistas y encuestas (Figura 5).

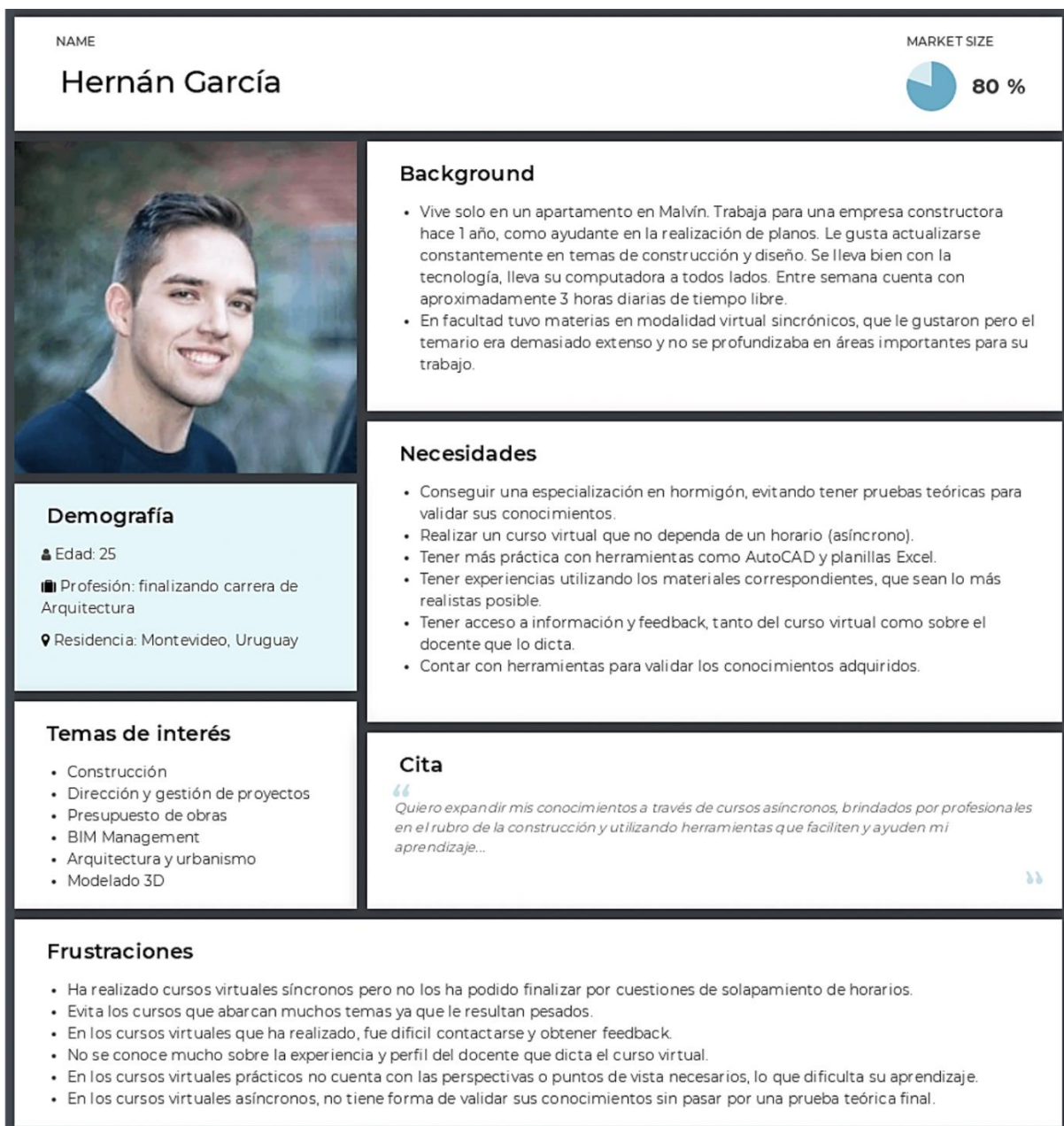


Figura 5 - User Persona

En primer lugar, es importante destacar que todos los datos que se ilustran en la imagen se basan en la información e insights que obtuvimos a partir de las entrevistas y encuestas.

Incluimos datos demográficos tales como la edad, profesión y lugar de residencia. En este caso, nuestro User Persona es un estudiante avanzado de Arquitectura, relativamente nuevo en el ámbito laboral, con poca experiencia y que busca constantemente mantenerse al día en temas relacionados al rubro de la construcción, algo que es muy importante para nosotros debido a que nuestros usuarios deben estar interesados en seguir aprendiendo y actualizándose en el área.

El hecho de que se lleve bien con la tecnología es importante ya que nuestra plataforma está basada en ello.

Las horas libres diarias con las que cuenta son pocas, por lo que es importante que los cursos se adapten a sus horarios, pudiéndolos ver cuándo y dónde quiera.

Las necesidades representan una serie de puntos que pretendemos cubrir, y que hoy en día, con las opciones disponibles, no es posible cumplirlas.

Los temas de interés están basados en las respuestas que obtuvimos de las encuestas, colocando los más mencionados para poner algunos ejemplos de posibles temas.

También agregamos una cita, la cual representa de forma más genérica a nuestros usuarios; personas que quieran expandir sus conocimientos a través de cursos virtuales asíncronos, brindados por profesionales del rubro de la construcción, a la vez que se utilizan las herramientas más relevantes y actuales del sector.

Como última sección, mencionamos frustraciones basadas en lo que los entrevistados nos comentaron, así como también en las encuestas, y que notamos que se repetían en varias ocasiones.

4.1.4. Benchmarking

Otro punto que consideramos fue el de nuestros competidores, o aquellos que podrían llegar a serlo en un futuro. Para esto, realizamos un análisis de ciertos aspectos de los mismos, concluyendo lo que explicamos a continuación:

- **Estructura de la competencia: madura**

Consideramos que los competidores existentes ya se encuentran establecidos y llevan cierto tiempo en el mercado, por lo que su estructura es madura.

- **Número de competidores directos: pocos**

La cantidad de entidades que brindan cursos enfocados al rubro de la construcción es relativamente baja, teniendo en cuenta la demanda. Entre ellos encontramos a Escuela Ma-pa, CECATEC, INEFOP y FOCAP.

Algunos cuentan con el apoyo del estado y tienen bastante tiempo en el rubro. Sin embargo, presentan la problemática de enfoques totalmente presenciales y pocas herramientas para brindar a los estudiantes la motivación necesaria para capacitarse y llevar a la práctica lo aprendido en los cursos.

Si también consideramos lo internacional, encontramos a la plataforma de aprendizaje Udemy. A pesar de contar con algunos cursos relacionados a la construcción, la mayoría están en inglés, carecen de enfoques en cuanto a lo práctico y sus certificaciones no tienen validez legal, ya que no se trata de una institución acreditada.

- **Número de competidores indirectos: muchos**

Como competidores indirectos tenemos a los centros educativos, como pueden ser las universidades, UTU y a YouTube. No obstante, este último no cuenta con ningún tipo de respaldo o certificación en cuanto a la fidelidad y perfil de las personas que brindan contenido, ya que se trata de una plataforma de uso libre y gratuito. Esto tiene como consecuencia la pérdida de tiempo al realizar búsquedas y que las personas pueden adquirir conocimientos erróneos o que no son totalmente verídicos.

- **Número de sustitutos: ninguno**

No encontramos competidores sustitutos.

- **Competidores sigilosos: pocos**

Como ejemplo de esto, tenemos a Coursera, Domestika y Platzi. Todos ofrecen cursos virtuales, pero no se enfocan en el rubro de la construcción, ni hacen uso de las tecnologías y herramientas que nosotros proponemos.

- **Fuerza de competidores: media**

En base a toda la investigación anterior, la fuerza de nuestros competidores es media, ya que a pesar de que se encuentren establecidos en el mercado, brindan cursos sincrónicos con horarios fijos, cuentan con temas extensos y genéricos y pruebas teóricas obligatorias para la obtención de certificaciones. Esto representa un problema para aquellos trabajadores que quieren obtener conocimientos específicos, en el tiempo que tienen disponible y se adecua mejor a sus necesidades.

4.2. Análisis del mercado

Si bien en la sección anterior [4.1.4. Benchmarking](#) describimos puntos del mercado enfocados a los competidores, también analizamos otros aspectos del mismo, llegando a las conclusiones que comentamos a continuación.

En primer lugar, tenemos lo relacionado a las tendencias del mercado, explicado en los siguientes puntos:

- **Mercado macro: múltiple y convergente**

El mercado de e-learning es muy amplio y ocupa diversas áreas, tales como ciencia y tecnología, arte y entretenimiento, cocina, negocios, bienestar, entre otros. Debido a la pandemia reciente, el aprendizaje online se ha visto en crecimiento aún más, y las personas tuvieron que acostumbrarse y adaptarse a utilizar más la tecnología y comunicarse de manera virtual.

- **Mercado objetivo: múltiple y convergente**

Nuestro objetivo es el mercado que involucra el rubro de la construcción. Este mercado es amplio y cubre muchas áreas, como pueden ser lo relacionado al diseño y cálculo de estructuras, dirección de obras, diseño de caminería y pavimentos, vías de transporte, saneamiento y demás. Es por ello que consideramos que tenemos una muy buena oportunidad de formar parte de este mercado.

- **Ventana de oportunidad: abierta**

El aprendizaje online tiene mucho potencial, y la pandemia ha potenciado esto; las personas se han acostumbrado a comunicarse virtualmente y el uso de tecnologías también se ha visto en mucho más aumento. Por ello, creemos que la ventana de oportunidad para nuestro proyecto está abierta.

- **Estructura de mercado: emergente**

Si bien ya existen entidades que brindan cursos en el rubro de la construcción, siguen careciendo de aspectos como el ofrecer cursos virtuales asíncronos, fragmentación más específica de temas cubiertos en un solo curso, evitando así un temario extenso y genérico, así como información detallada y feedback acerca de los docentes que dictan dichos cursos. Es por ello que la estructura del mercado es emergente y se encuentra en constante crecimiento, lo cual representa una buena oportunidad para nuestro proyecto de insertarse en el mismo.

Luego, en cuanto a los aspectos relativos al tamaño del mercado, obtuvimos lo siguiente:

- **Grupos de clientes: núcleo grande**

Actualmente, existen alrededor de 8.000 arquitectos e ingenieros civiles recibidos y aproximadamente 11.000 estudiantes activos en dichas carreras. Estos números no incluyen a personas que no completaron estas carreras, pero igualmente trabajan en el rubro, graduados de carreras no universitarias, como pueden ser tecnicaturas, y personas con conocimientos avanzados de construcción. En resumen, estimamos un total de aproximadamente 45.000 a 50.000 personas. [4] [5]

Un dato no menor, es que, si hacemos este mismo análisis para Brasil, por ejemplo, solo contando a los arquitectos e ingenieros civiles activos, tendríamos

aproximadamente 570.000 personas. A esto habría que sumarle los estudiantes avanzados de estas carreras, técnicos constructores e individuos con conocimientos avanzados en el rubro. [6]

Esto también sucede, en proporción a la población, en otros países de Latinoamérica, como Perú, Argentina o Chile, donde nuestro servicio también podría verse aplicado.

- **Demanda: más grande que la oferta**

De acuerdo a nuestra encuesta realizada, de un total de 128 trabajadores del rubro de la construcción, un 84% ha realizado cursos virtuales, ya sea para capacitarse o actualizarse en temas relacionados al sector. Cada vez hay más personas que estudian estas carreras, por lo que los profesionales necesitan destacarse de algún modo y las opciones son acotadas, además de no adaptarse al 100% con las nuevas tecnologías.

Además, en base a nuestras entrevistas, muchas personas se inscriben a masters en el extranjero online, pero no los terminan ya que ocupan un temario extenso y solo quieren aprender sobre ciertos conocimientos específicos avanzados, que no se encuentran en cursos cortos.

Finalmente, concluimos lo siguiente acerca del crecimiento del mercado:

- **Tasa de crecimiento del mercado: entre 17.5% y 25%**

En base a nuestra investigación, las proyecciones indican un crecimiento del mercado que oscila entre el 17.5% y 25% del año 2021 al año 2029. [7] [8]

4.3. Público objetivo

Algo importante que queremos destacar es que, dado que nuestro proyecto es un emprendimiento, no contamos con un cliente concreto, sino que tenemos un público objetivo definido al cual enfocamos nuestro producto. En este caso, el mismo consta de trabajadores del rubro de la construcción, con cierta experiencia, técnicos o profesionales, que quieran adquirir nuevos conocimientos, especializarse o enseñar a otros.

4.4. Modelo de negocio Canvas

También realizamos una tabla de modelo de negocio Canvas, donde indicamos los puntos más relevantes para el desarrollo de nuestro producto (Figura 6).

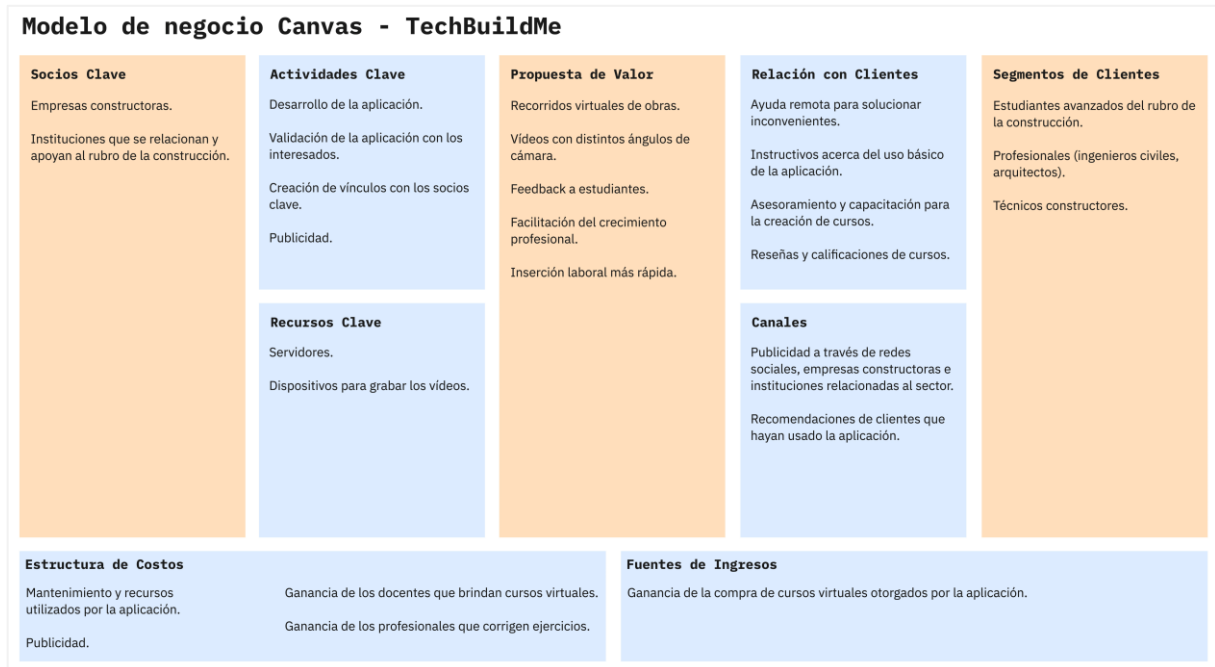


Figura 6 - Modelo de negocio Canvas para TechBuildMe

▪ Socios clave

Los convenios con empresas constructoras e instituciones relacionadas al rubro brindarán credibilidad y confianza a nuestros clientes, respaldando el contenido ofrecido por nuestra plataforma, así como también validando los certificados otorgados por la misma. Además, dichos convenios ayudarán a la realización de recorridos virtuales de obras en las instalaciones de las empresas.

▪ Actividades clave

Nuestras actividades esenciales incluyen el darle foco al desarrollo de la plataforma, realizando validaciones de forma continua con los interesados de modo que podamos recolectar opiniones acerca de su impacto en el rubro y posibles mejoras.

El crear y mantener buenas relaciones con nuestros socios clave también resulta de suma importancia, ya que, como mencionamos anteriormente, tener convenios con ellos resultaría en un gran beneficio para el crecimiento de nuestro producto. Lo mismo

sucede con la publicidad, la cual nos ayudaría a alcanzar mayor cantidad de público objetivo a través de diversos medios.

- **Recursos clave**

Tener servidores para guardar la información de los usuarios y datos que puedan resultar relevantes para la plataforma es importante ya que nos permitiría una obtención más rápida de los mismos, además de ser una forma de respaldo.

Otro recurso clave son los dispositivos para realizar las grabaciones de los videos, como son las cámaras de 360 grados para los recorridos virtuales en obras.

- **Propuesta de valor**

Consideramos que nuestra propuesta de valor se centra en ofrecer recorridos virtuales de obras, ya que es lo que creemos que va a aportar un diferencial al público objetivo. Esto debido a que ofrecemos la oportunidad de poder visitar obras de construcción de forma virtual, haciendo la experiencia lo más inmersiva y realista posible, sin tener que desplazarse físicamente hasta el lugar, lo que resulta en ahorro de tiempo y gasto por parte del usuario.

El ofrecer distintos ángulos de cámara para un video y brindar feedback a los estudiantes también agrega valor al producto, ya que impacta en el aprendizaje de los mismos. A través de estas herramientas, mejoramos y facilitamos la forma en que adquieren conocimientos, ayudando a su crecimiento profesional e inserción laboral.

- **Relación con clientes**

En cuanto al relacionamiento con los clientes, queremos brindarles ayuda en caso de tener inconvenientes con la plataforma, así como instructivos de uso básico para guiarse a través de la misma.

También ofreceríamos asesoramiento para aquellos usuarios que quieran compartir sus conocimientos a través de la creación de cursos en la plataforma. Asimismo, permitiremos a los clientes puntuar los cursos que realizan para obtener sus opiniones y posibles mejoras que podamos realizar para enriquecer aún más su experiencia con nuestro producto.

- **Segmentos de clientes**

Son trabajadores del rubro de la construcción, ya sea estudiantes avanzados, profesionales (como ingenieros civiles y arquitectos) y técnicos constructores.

- **Canales**

Como canal de comunicación con nuestros clientes, tenemos el hacer publicidad a través de redes sociales, empresas constructoras e instituciones que se relacionen con el rubro de la construcción. Otro canal son las recomendaciones de clientes que hayan probado la plataforma a personas del rubro que conozcan.

- **Estructura de costos**

Los principales costos serán los relacionados al mantenimiento y recursos utilizados por la plataforma, así como la ganancia de los docentes que brindan cursos virtuales y la de los profesionales que corrijan ejercicios, descontándola del costo total del curso. La publicidad también puede generar costos ya que podríamos tener que contratar a algún tercero que se encargue de ello.

- **Fuentes de ingresos**

Las ganancias obtenidas por la compra de los cursos virtuales otorgados por nuestro producto por parte de los clientes será nuestra fuente de ingreso.

4.5. Nombre y logo

Luego de discutir una lista de posibles nombres, optamos por algo que fuera intuitivo y que contuviera palabras relacionadas al rubro y lo que queríamos lograr con la plataforma; capacitar o “construir” técnicamente a nuestros clientes para ayudar a su crecimiento en el ámbito laboral. Es por ello que decidimos nombrar a nuestra plataforma como TechBuildMe, también pudiendo abreviarse como TBM.

En cuanto al logo, pensamos en algo que no fuera demasiado complejo, legible y simple como puede apreciarse a continuación (Figura 7).



Figura 7 - Logo de la aplicación

A modo de comentario, esta y las anteriores secciones nos permitieron elaborar la documentación requerida para la presentación formal de nuestro producto al CIE. Para ver en detalle el documento ir a [13.3. Anexo 3: Documentación CIE](#).

5. Marco metodológico

En este capítulo, describiremos el uso de las metodologías utilizadas para la construcción del proyecto, usando como base las características de un proyecto emprendedor, mezcladas con los requerimientos establecidos a nivel académico.

A su vez, detallaremos las características de las metodologías, los roles y las conclusiones del uso de dichas metodologías en un proyecto de estas particularidades.

5.1. Características del proyecto

En esta sección detallaremos particularidades de nuestro proyecto, donde muchas de las decisiones tomadas se basan en el hecho de estar iniciando un emprendimiento.

En primer lugar, queremos destacar que la definición de la idea inicial comenzó en enero de 2022. Allí comenzamos a investigar más acerca de las necesidades que podía tener el sector de la construcción y, posteriormente, comenzamos una etapa de relevamiento de datos, cuyos resultados comentamos en el capítulo [4. Emprendimiento](#) de este documento. Con la ayuda de estos datos obtenidos, fue que afinamos la idea y la presentamos al CIE, para poder obtener una aprobación del emprendimiento por su lado.

Una vez que comenzamos el proyecto formalmente, tuvimos en cuenta ciertos puntos que tendrían un impacto en el desarrollo del mismo, propios de un emprendimiento.

- **Cliente real no definido**

Como ya hemos mencionado anteriormente, no contamos con un cliente establecido, sino con un público objetivo dentro del rubro de la construcción, en el cual nos enfocamos para realizar el proyecto. Esto resultó en que tuvimos que definir los requerimientos y el alcance nosotros mismos, en base a la extensa investigación previa que llevamos a cabo. A su vez, tuvimos que buscar interesados que estuvieran dispuestos a realizar sesiones periódicas de validación de nuestro producto, de modo de poder obtener feedback y posibles mejoras.

- **Validaciones recurrentes con los interesados**

En base a lo comentado en el punto anterior, el tener sesiones de validación de forma recurrente es muy importante para nuestro proyecto. Estas instancias nos permiten probar y mostrar el producto a los interesados, los cuales representan parte del público objetivo, obteniendo sus opiniones acerca del mismo. Estas pueden incluir comentarios sobre aspectos que podrían ser mejorados, ayudándonos a poder optimizar nuestro producto de forma que aporte el mayor valor posible al sector.

- **El producto depende de los clientes**

Nuestra plataforma se basa en brindar contenido para que nuestro público objetivo pueda capacitarse o especializarse en temas relativos al sector. Por este motivo, dependemos mucho de dicho público, en especial de los que tienen el rol de docente, ya que son los que tienen experiencia en el sector y pueden crear contenido en nuestra plataforma e impartir sus conocimientos a otros.

- **Modificaciones en lo planificado**

Por lo detallado anteriormente, entendíamos que a medida que el proyecto fuera avanzando, tendríamos que afrontar diferentes cambios. De hecho, los requerimientos y el alcance iniciales del proyecto sufrieron cambios debido a resultados que obtuvimos de etapas de validación con los interesados, como comentaremos en secciones más adelante.

5.2. Características del equipo

En esta sección, detallaremos las cualidades que el equipo posee y que tuvieron impacto en la forma en que abordamos el proyecto.

- **Experiencia previa trabajando en conjunto durante la carrera**

El hecho de que veníamos realizando materias en conjunto desde el segundo semestre de la carrera nos ayudó a organizarnos mejor, ya que conocemos las fortalezas y disponibilidad de cada uno, pudiendo complementarnos y repartir las tareas de forma más eficiente. Asimismo, esto contribuyó a tener una buena

comunicación y confianza entre nosotros, pudiendo discutir al momento de tomar decisiones acerca del proyecto de manera más directa y honesta.

- **Experiencia laboral en desarrollo de software**

Ambos tenemos experiencia trabajando en empresas como desarrolladores de software, lo que nos facilitó la elección y el manejo de las tecnologías a utilizar en el proyecto. A su vez, diariamente ponemos en práctica metodologías de trabajo, con ciertas adaptaciones, lo que también ayudó a escogerlas para nuestro proyecto.

- **Primera experiencia con un emprendimiento**

Ninguno de los dos ha llevado adelante ningún tipo de emprendimiento, por lo que este proyecto representó ciertos desafíos y dificultades que fuimos abordando de modo de que repercutiera lo menor posible en lo que teníamos planificado.

5.3. Metodologías de trabajo

5.3.1. Lean Startup y Scrum

Como comentamos anteriormente, debido a que este es un proyecto emprendedor, no tenemos un cliente real definido. En su lugar, tenemos un público objetivo al que nos estamos enfocando. Es por esto que tener etapas constantes donde validamos nuestro producto con esas personas fue muy importante para el desarrollo del mismo, ya que no contábamos con requerimientos predefinidos ni expertos en el tema de forma directa. Como consecuencia de esto, también tuvimos que invertir mucho tiempo en investigación en las primeras etapas del proyecto, así como ser precavidos a la hora de tomar de ciertas decisiones relacionadas con modificaciones en lo planificado. De esta forma, buscamos reducir el riesgo de impacto lo más posible, también teniendo en cuenta que éramos solamente dos personas y debíamos organizar nuestros tiempos de la forma más eficiente.

Por todo lo anterior, al momento de seleccionar las metodologías que íbamos a usar, entendimos importante el alineamiento con las validaciones y los cambios constantes producto de dichas validaciones. Es por esto que elegimos Lean Startup, ya que sus características definen claramente un proceso de construcción (sin especificar el

“cómo”) un proceso de medición, donde se valida lo construido, y una etapa de aprendizaje, a partir de la cual se obtienen mejoras.

A pesar de que Lean Startup se adapta a nuestro proyecto, esta metodología no define "cómo" se tiene que llevar a cabo la etapa de construcción, la cual consta de un proceso sumamente complejo. Por este motivo, notamos que necesitábamos una metodología que nos ayudara con ello, que nos permitiera agregar cambios durante el transcurso del desarrollo, mantenernos en comunicación en cuanto a tareas y definir cómo íbamos a organizar el trabajo.

En consecuencia, seleccionamos SCRUM ya que la consideramos la más adecuada para nuestro proyecto. Esto debido a que, mediante el uso de Sprints, nos permitió desarrollar y tomar pausas para analizar el estado del Sprint actual y qué acciones podíamos tomar para el siguiente, por medio de Sprint Reviews y Sprint Retrospectives. Además, durante el transcurso de la etapa de construcción fue posible tomar o agregar tareas del Backlog, estando en comunicación constantemente acerca nuestro progreso a través de una Daily interna.

En adición, el hecho de planificar cada Sprint con antelación (Sprint Planning) con las métricas tomadas de reuniones que íbamos realizando, nos ayudó a conocer cada vez más nuestra velocidad de trabajo y proyectar la duración de los ciclos de forma más exacta.

Por todo lo mencionado anteriormente, estas dos metodologías juntas, sumado a la modificación de algunos aspectos preestablecidos de las mismas (que mencionaremos en la siguiente sección [5.3.2. Aplicación de las metodologías](#)) con el fin de adecuarse más exactamente a nuestro proyecto y forma de trabajo, nos facilitaron el cumplimiento de los objetivos del proyecto.

5.3.2. Aplicación de las metodologías

En esta sección, detallaremos como aplicamos las metodologías Lean Startup y SCRUM, utilizadas en conjunto, con las modificaciones realizadas con el fin de adaptarse a las características de nuestro proyecto.

A continuación, mostramos un diagrama representando el flujo de trabajo que llevamos adelante en base a ambas metodologías (Figura 8).

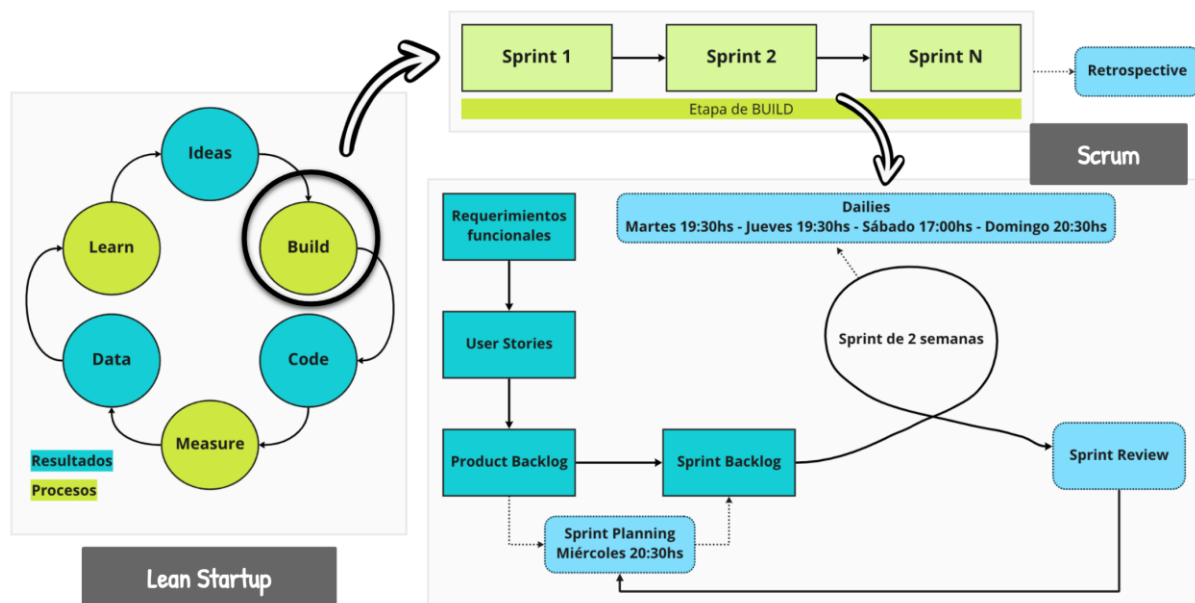


Figura 8 - Diagrama de las metodologías Lean Startup y SCRUM

El ciclo de Lean Startup consta de ciclos cortos, partiendo desde las ideas obtenidas a partir del análisis de las validaciones que se transforman en requerimientos funcionales (RF). Sin embargo, en nuestro caso no fue plenamente así debido a que nuestro proyecto es un emprendimiento y a la vez un proyecto académico, por lo que contábamos con un tiempo limitado y al momento de alcanzar el plazo de entrega definido, debía tener el valor de un proyecto de fin de carrera. Para asegurar esto, fue necesario definir de antemano todos los RF, lo cual no se encuentra alineado completamente con Lean Startup, ya que estos RF deberían ir surgiendo como resultado de cada ciclo corto.

No obstante, pese a que conocíamos los RF a realizar en los próximos ciclos, tuvimos instancias de validación con los interesados correspondientes al proceso de medición de Lean Startup, arrojando datos, los cuales luego de un análisis (equivalente al proceso de aprendizaje) generaron nuevas ideas o posibles mejoras a aplicar a nuestro producto en el próximo ciclo.

En el proceso de construcción es donde aplicamos SCRUM para definir el “como” íbamos a llevarlo a cabo. Para ello, una vez definidos los requerimientos funcionales, se registraron en forma de Epics, las cuales se desglosaron en User Stories (US), que a su vez, se fragmentaron en tareas más pequeñas con el fin de poder paralelizar el trabajo.

Para poder decidir que US eran las adecuadas para trabajar en cada Sprint (que duraban dos semanas) y medir nuestro progreso a medida que terminábamos uno, realizamos distintas ceremonias que mencionaremos a continuación.

Antes de comenzar un Sprint, teníamos una Sprint Planning, donde analizábamos las tareas por hacer y, según lo que representaba ese Sprint, les asignábamos prioridad y las agregábamos al Sprint Backlog. Luego, durante el transcurso del Sprint, realizábamos Dailies cuatro veces a la semana para ponernos al día acerca de cuál era nuestra situación actual y si teníamos algún bloqueo o no. El motivo por el cual no realizamos esta ceremonia todos los días se debió a que, al ser dos personas, estábamos en constante comunicación para conocer nuestros estados actuales y por si surgía algún inconveniente, por lo que decidimos dejar un espacio para poder enfocarnos en lo que teníamos asignado y avanzar más rápido.

Al finalizar cada Sprint, teníamos una Sprint Review para analizar cómo nos había ido, si debíamos mejorar aspectos relativos al manejo de las tareas, como podían ser estimaciones o reparticiones de las mismas. Seguidamente, debido a la corta duración de esta ceremonia, volvíamos a realizar una Sprint Planning del siguiente Sprint.

Una vez que finalizábamos una serie de Sprints y efectuábamos un Release (del cual hablaremos más en detalle en la sección [8.4. Plan de releases](#)) llevamos a cabo una Sprint Retrospective, donde analizábamos que estábamos haciendo bien, que debíamos mejorar y que debíamos descartar o cambiar completamente.

Posteriormente, obteníamos un Release funcional, correspondiente al resultado del código del proceso de construcción de Lean Startup, que podía ser testeado y validado y que reflejaba lo que habíamos logrado en las etapas anteriores. Luego realizábamos un análisis de cumplimiento de los criterios de aceptación de cada User Story involucrada en el Release, generando tareas que representaban bugs en caso de no cumplirse alguno. También llevábamos a cabo validaciones con los interesados, lo que representaba el proceso de medición, de las cuales obteníamos información útil para analizar en el proceso de aprendizaje, con el fin de conseguir posibles mejoras. De esta forma, con las ideas obtenidas, comenzaba nuevamente el ciclo, a partir del proceso de construcción.

Como punto a destacar, era posible que la medición no se finalizara y que igualmente comenzáramos con el siguiente proceso de construcción. Sin embargo, al terminarse el proceso de construcción siguiente, si debía estar terminado el proceso de medición y aprendizaje anterior.

A modo de comentario, no se tuvieron en cuenta los roles definidos por SCRUM dado que, al ser dos integrantes, ambos podíamos cumplir todos los roles y dicha especificación no aportaba valor significativo.

5.3.3. Ciclo de vida

El ciclo de vida definido de nuestro proyecto es híbrido, siendo una combinación de un modelo adaptativo y uno iterativo e incremental.

En las primeras etapas comenzamos con un modelo adaptativo; tuvimos una etapa extensa de relevamiento de datos, contacto con el público objetivo para obtener insights, validaciones de la idea inicial a través de prototipos, definición de RF y RNF y definición del diseño arquitectónico. Estas etapas iniciales tendían a cambiar mucho ya que buscábamos empatizar continuamente con el público objetivo (para más detalle ver [4.3. Público objetivo](#)), a la vez que validábamos las necesidades del sector.

Posteriormente, cuando hicimos una refinación de la idea inicial según los resultados que fuimos obteniendo, el modelo también se inclinó hacia un modelo iterativo e incremental. Esto debido a que al obtener un Release, se validaba con los interesados y, en base a su feedback, creábamos una US reflejando el ajuste correspondiente, pudiéndolo tener en cuenta para el próximo Release.

Aunque las características de un emprendimiento no se ajustan totalmente con un modelo iterativo e incremental, consideramos que esta fue la mejor forma de planificar teniendo en cuenta la duración del proyecto. Al ser de tipo académico, dicha duración es un punto constante que no podemos modificar, por lo que debíamos asegurar el cumplimiento de lo que planeamos dentro del tiempo límite del proyecto, estableciendo previamente los requerimientos.

5.4. Roles

Debido a que el equipo solo consta de dos integrantes, no tuvimos roles asignados específicamente, ya que estábamos en constante comunicación para estar al tanto de nuestro progreso y para organizar las tareas y demás. Para la repartición de las mismas, tuvimos en cuenta las destrezas de cada uno para poder ser lo más eficientes y equilibrados posible.

5.5. Conclusiones de las metodologías usadas

Lo que podemos afirmar en base al uso de Lean Startup y SCRUM, es que nos fueron útiles para el desarrollo de este tipo de proyecto emprendedor, teniendo en cuenta las modificaciones realizadas, dado que el proyecto tiene un alcance limitado, por lo que no sería posible seguir completamente los lineamientos que establecen las metodologías.

A su vez, el uso de estas metodologías nos resultó en un beneficio ya que ambos comprendimos como aplicarlas y adaptarlas a un proyecto emprendedor. De esta manera, en una futura etapa posterior a la entrega del proyecto, nos ayudará a mantener una gestión más eficiente y ordenada de nuestro emprendimiento y actividades profesionales.

En las próximas secciones, se podrá ver en detalle como el uso de estas metodologías nos permitieron el desarrollo del producto.

6. Ingeniería de requerimientos

En este capítulo, detallaremos el proceso que llevamos a cabo para la obtención de los requerimientos funcionales (RF) y no funcionales (RNF) del producto. Dentro de esto, se incluyen los insights y los prototipos realizados a partir de estos y el refinamiento realizado para reducir el alcance de los RF, así como la descomposición de estos requerimientos en User Stories.

A su vez, se mencionarán los criterios de aceptación para los RNF, la definición de nuestro MVP y los RF extra que fueron creados en base a nuestra planificación.

6.1. Proceso de definición de requerimientos

6.1.1. Elicitación

Como primer paso para poder definir los requerimientos, realizamos un relevamiento de datos, detallado en la sección [4.1. Relevamiento de datos](#).

Una vez obtenidos los resultados de las entrevistas y encuestas a trabajadores del rubro de la construcción, realizamos un análisis de los mismos, lo que nos permitió conseguir un conjunto de insights, descriptos en la figura a continuación (Figura 9).

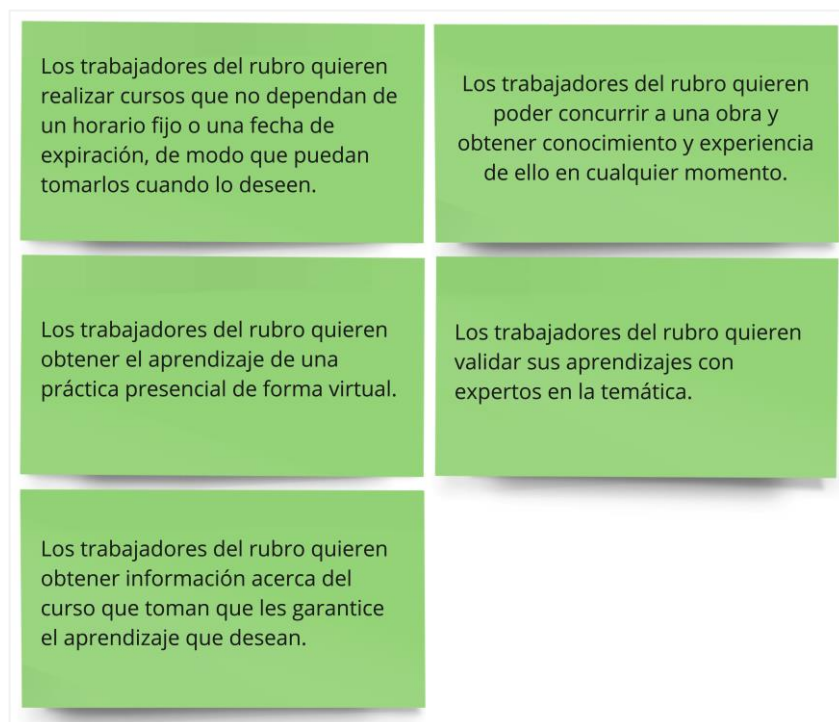


Figura 9 - Insights obtenidos

En base a estos insights, hicimos un análisis de las necesidades y aspectos que considerábamos que carecía el mercado, elaborando así una serie de prototipos de la plataforma (para más detalle ver [13.4. Anexo 4: Prototipos realizados](#)) con el objetivo de obtener feedback y realizar validaciones de los mismos con personas que pertenecieran a nuestro público objetivo (detallado en [4.3. Público objetivo](#)).

Para las instancias de validación de prototipos, nos acompañaron todos los interesados mencionados en detalle en la sección [2.4. Interesados que nos acompañaron](#).

Luego de validar los prototipos con los interesados y ajustarlos en base a sus comentarios, rectificando que las necesidades del sector estaban alineadas con nuestros insights, utilizamos dichos prototipos para definir un total de 13 requerimientos funcionales (RF), como puede apreciarse en la Figura 10.

RF1 - Creación de cursos virtuales, incluyendo subida de videos y archivos.	RF2 - Visualización de videos con opción de ver distintos ángulos de cámara simultáneamente.	RF3 - Búsqueda, catálogo de cursos agrupados por temas y recomendaciones en base a gustos y preferencias del usuario.	RF4 - Selección de cursos virtuales como favoritos.
RF5 - Recorridos virtuales de obras.	RF6 - Perfil de usuario, incluyendo nombre, apellido, formación, situación laboral, foto, descripción (parte de la información podrá ser opcional).	RF7 - Subida y visualización de ejercicios entregables.	RF8 - Corrección y feedback de ejercicios realizados por expertos en la temática.
RF9 - Manejo de tokens internos como forma de recompensas a docentes que corrijan trabajos.	RF10 - Canal de comunicación directa con los docentes.	RF11 - Compra de cursos virtuales.	RF12 - Soporte manejado principalmente por un bot, de modo que los usuarios puedan resolver cualquier inconveniente que puedan tener.
RF13 - Canal con ofertas laborales de empresas constructoras.			

Figura 10 - Lista de RF iniciales

6.1.2. Priorización y reescritura

Una vez definidos los RF mencionados anteriormente, procedimos a reunirnos con los interesados para validarlos, mostrándoles la relación con los prototipos anteriores, y pedirles que les asignaran prioridad alta, media o baja. Aquellos con prioridad alta eran esenciales y por lo tanto debían estar presentes en nuestro producto, mientras que el resto, con media o baja, los consideramos como deseables (Figura 11).

RF1 - Creación de cursos virtuales, incluyendo subida de videos y archivos. PRIORIDAD MEDIA	RF2 - Visualización de videos con opción de ver distintos ángulos de cámara simultáneamente. PRIORIDAD ALTA	RF3 - Búsqueda, catálogo de cursos agrupados por temas y recomendaciones en base a gustos y preferencias del usuario. PRIORIDAD MEDIA	RF4 - Selección de cursos virtuales como favoritos. PRIORIDAD MEDIA
RF5 - Recorridos virtuales de obras. PRIORIDAD ALTA	RF6 - Perfil de usuario, incluyendo nombre, apellido, formación, situación laboral, foto, descripción (parte de la información podrá ser opcional). PRIORIDAD BAJA	RF7 - Subida y visualización de ejercicios entregables. PRIORIDAD ALTA	RF8 - Corrección y feedback de ejercicios realizados por expertos en la temática. PRIORIDAD ALTA
RF9 - Manejo de tokens internos como forma de recompensas a docentes que corrijan trabajos. PRIORIDAD BAJA	RF10 - Canal de comunicación directa con los docentes. PRIORIDAD MEDIA	RF11 - Compra de cursos virtuales. PRIORIDAD BAJA	RF12 - Soporte manejado principalmente por un bot, de modo que los usuarios puedan resolver cualquier inconveniente que puedan tener. PRIORIDAD MEDIA
RF13 - Canal con ofertas laborales de empresas constructoras. PRIORIDAD BAJA			

Figura 11 - Lista de RF iniciales priorizados

Los mismos pueden verse, junto con su escritura en formato de narrativa, en la sección [13.5. Anexo 5: RF iniciales](#).

Sin embargo, luego de que realizáramos nuestra planificación, teniendo en cuenta el tiempo con el que contábamos, la complejidad del conjunto de RF definidos y que éramos un equipo de dos personas, notamos que el alcance era muy amplio como para cumplir con todo lo que pretendíamos dentro del tiempo límite del proyecto. Además, había ciertos RF que no eran realmente prioritarios en cuanto al aporte de valor que podrían agregar a nuestro producto.

Por lo tanto, considerando la prioridad asignada a cada RF inicial por nuestros interesados, hicimos una reevaluación y revisión de los mismos, para luego realizar una reescritura de los que poseían una prioridad alta. Fue así como terminamos definiendo cuatro RF principales, que englobaban las funcionalidades que realmente eran importantes y que aportaban valor a nuestro producto.

Una vez hecho esto, volvimos a reunirnos con los interesados para validar este conjunto de RF, mostrándoles la relación entre estos y los prototipos actualizados, con el fin de confirmar con estos individuos que la cohesión entre estos RF era alta y que eran suficientes para marcar un diferencial en el sector.

Esta instancia de validación nos fue de gran ayuda al momento de definir nuestro MVP, del cual hablaremos más en detalle en la sección [6.4. Definición de MVP](#), así como para generar un plan de Releases para gestionar la etapa de desarrollo, el cual puede visualizarse en la sección [8.4. Plan de Releases](#).

Asimismo, en base a dicho plan de Releases y al tiempo restante de desarrollo con que disponíamos, decidimos agregar dos RF no prioritarios extras al alcance inicial, los cuales los redefinimos a partir del conjunto de los RF iniciales que tenían una prioridad media y eran un complemento a las cuatro funcionalidades principales. Estos también fueron validados con los interesados de la misma forma que los cuatro RF principales.

En la siguiente imagen (Figura 12) se pueden visualizar los seis RF finales, los cuales serán explicados en detalle en la sección [6.2. Requerimientos funcionales](#).



Figura 12 - RF finales

6.1.3. Especificación

Luego de relevar y redefinir los RF, procedimos a dividirlos en US, siguiendo un formato incluyendo título, prioridad, estimación, cuerpo de la User Story y criterio de aceptación en forma de escenarios (Figura 13).

Título:	Estimación:
User Story: Como [descripción del usuario], Quiero [funcionalidad], De modo que [beneficio].	
Criterio de aceptación: Dado [comienzo de las cosas] Cuando [acción tomada] Entonces [resultado de aplicar la acción]	

Figura 13 - Formato de una User Story

La idea detrás de este formato era que cada funcionalidad se viera reflejada de forma breve y concreta desde el punto de vista del usuario, para mejor entendimiento de la misma y su valor aportado. Luego, creamos tareas dentro de cada una de estas US, las cuales fueron repartidas entre el equipo para ser llevadas a cabo.

A continuación, mostraremos el resultado de desglosar nuestros RF en US (Tabla 4). La nomenclatura utilizada para las US consta de TBMJIRA-[número autoasignado por Jira].

RF1 – Cursos	
ID	Nombre
TBMJIRA-8	Videos de un curso
TBMJIRA-18	División en módulos de un curso
TBMJIRA-28	Distintos ángulos de cámara de un curso
TBMJIRA-30	Vista previa de un curso
TBMJIRA-34	Sección de preguntas de un curso
TBMJIRA-38	Archivos informativos de un curso

RF2 – Recorridos virtuales	
ID	Nombre
TBMJIRA-42	Implementación de recorrido virtual usando video 360
TBMJIRA-49	Preguntas a lo largo del recorrido virtual
TBMJIRA-58	Sección de explicación de funcionamiento del recorrido virtual
RF3 – Ejercicios entregables	
ID	Nombre
TBMJIRA-69	Registro de un usuario
TBMJIRA-73	Entrega de ejercicios entregables
TBMJIRA-77	Detalle de ejercicio entregable de un curso
TBMJIRA-86	Visualización de consigna de ejercicio en lección
TBMJIRA-89	Descarga de ejercicio entregado
TBMJIRA-92	Listado de ejercicios entregables
RF4 – Feedback	
ID	Nombre
TBMJIRA-102	Listado de ejercicios entregados sin revisar
TBMJIRA-107	Visualización de una entrega
TBMJIRA-113	Agregar comentarios a una entrega
TBMJIRA-117	Visualización de comentarios en un ejercicio
RF5 – Favoritos	
ID	Nombre
TBMJIRA-126	Búsqueda de un curso
TBMJIRA-132	Selección de un curso como favorito
TBMJIRA-136	Listado de cursos favoritos
RF6 – Creación de curso	
ID	Nombre
TBMJIRA-147	Creación de curso nuevo
TBMJIRA-151	Listado de cursos creados
TBMJIRA-155	Modificación de curso (admin)
TBMJIRA-159	Solicitud de creación de curso
TBMJIRA-163	Listado de solicitudes de creación de curso (admin)

Tabla 4 - Desglose de RF en US

Para ver en detalle las US ir a [13.6. Anexo 6: Desglose de US](#).

6.2. Requerimientos funcionales

En esta sección listaremos en detalle las funcionalidades que forman parte del alcance del proyecto, presentándolas de mayor a menor prioridad.

▪ **RF1 – Cursos**

El sistema debe permitir la visualización de cursos virtuales con sus respectivos módulos, conteniendo videos (con distintos ángulos de cámara, opcionales) y archivos correspondientes. Los videos deben ser visualizados en alta calidad, es decir, full HD 1920x1080 pixeles para los videos con varios ángulos de cámara y 2048x124 para los videos 360 de recorridos virtuales.

▪ **RF2 – Recorridos virtuales**

El sistema deberá permitir, como parte de un curso, visualizar videos mediante el uso de una herramienta de realidad virtual que permita a los usuarios recorrer una obra.

Involucra hacer que nuestra plataforma sea compatible con videos 360, permitiendo al usuario desplazar la cámara en todas direcciones. Asimismo, poder visualizar preguntas relacionadas al curso durante el recorrido para practicar lo aprendido, haciendo la experiencia más interactiva, a la vez que se asemeja lo más posible a la realidad.

▪ **RF3 – Ejercicios entregables**

El sistema debe permitir la realización de ejercicios a través de la subida de archivos en los formatos pdf y zip.

Estos formatos permiten subir una documentación básica en pdf o, en el caso de que se requiera entregar más de un archivo, se podrán comprimir en un zip, puesto que es un formato universal de compresión.

▪ **RF4 – Feedback**

El sistema debe permitir a los docentes encargados de cursos, corregir y brindar feedback de los ejercicios entregados por los estudiantes, pudiendo entregar en la

devolución comentarios. Los comentarios pueden ser relativos al curso y se podrá agregar un link opcional a información relevante que sirva de apoyo al feedback otorgado.

- **RF5 – Favoritos**

El sistema debe permitir a un usuario buscar un curso y marcarlo como favorito (siempre y cuando no sea creado por él). A su vez, podrá ver el listado de sus cursos favoritos.

La búsqueda puede ser por nombre, docente o descripción de un curso. Los resultados se presentan en orden de relevancia, con aquellos cursos que contengan el texto de búsqueda en el nombre en primer lugar, seguidos de los cursos cuyo docente coincida con el término de búsqueda y, finalmente, los cursos que tengan palabras coincidentes en su descripción.

- **RF6 – Creación de curso**

El sistema debe permitir a un usuario de tipo administrador la creación de cursos virtuales, incluyendo título, descripción y creación de módulos, conteniendo dentro lecciones. Estas últimas pueden ser ejercicios entregables, material informativo, recorridos virtuales o videos, los cuales pueden ser subidos a la plataforma desde el ordenador.

El progreso del armado del curso debe quedar guardado hasta su finalización. Una vez creado, se podrá enviar el curso para que sea publicado en la plataforma. A su vez, se permitirá la modificación del contenido de estos cursos.

Los usuarios no administradores podrán enviar solicitudes para crear un curso, enviando una descripción del mismo e información personal que respalde sus conocimientos y formación en el sector. Esto con el fin de que los usuarios administradores puedan ver este listado de solicitudes y contactarse con estos usuarios para revisar la solicitud y posteriormente, comenzar la creación del curso.

6.3. Requerimientos no funcionales

A continuación, detallamos los RNF, agrupándolos por atributo de calidad.

- **Disponibilidad**

- RNF1 – El sistema debe tener una disponibilidad del 95% de las veces en que un usuario intente accederlo.

- **Eficiencia**

- RNF2 – El sistema deberá poder soportar 100 usuarios visualizando videos de forma simultánea.
- RNF3 – El sistema deberá manejar de forma eficiente el procesamiento de vídeo.

- **Seguridad**

- RNF4 – Se deberá proveer mecanismos de seguridad que garanticen la integridad de los archivos que contiene un curso.

- **Usabilidad**

- RNF5 – Cumplimiento de las heurísticas de Nielsen.
- RNF6 – La plataforma debe ser responsive a fin de garantizar la adecuada visualización en múltiples dispositivos.
- RNF7 – La plataforma debe seguir un sistema de diseño (Material Design).

- **Portabilidad**

- RNF8 – La plataforma deberá ser compatible con los navegadores Google Chrome y Safari, en sus versiones de escritorio y, en el caso de móviles, con sistema operativo iOS compatibles con Safari y Android compatibles con Google Chrome.

Cada uno de estos RNF tiene definido un criterio de aceptación que se ven a continuación (Tabla 5), para los cuales proveeremos evidencia de su cumplimiento en los capítulos [7. Arquitectura y Diseño](#) y [10. Gestión de la calidad](#).

RNF #	Criterios de aceptación
RNF1	La disponibilidad del sistema debe tener una disponibilidad no menor al 95%
RNF2	Se deberá brindar evidencia o justificación de configuración que demuestre que la plataforma soporta 100 usuarios reproduciendo video en simultaneo.
RNF3	Se deberá brindar evidencia o justificación que permita demostrar que se realizó un manejo eficiente del procesamiento de los videos.
RNF4	Se deberá brindar evidencia de mecanismos utilizados para salvaguardar la integridad de los archivos de un curso.
RNF5	Se deberá brindar capturas que evidencien el cumplimiento de las mismas o si fuera el caso, justificar porque una heurística no se cumple.
RNF6	El sitio de debe poder ajustar a los distintos tamaños de pantalla, manteniéndose responsive hasta un ancho de 400 pixeles.
RNF7	Se deberá proveer capturas y justificación que demuestren el uso de un sistema de diseño en el sitio.
RNF8	Se deberá poder acceder a la web app por medio de los navegadores Google Chrome y Safari, en sus versiones de escritorio y, en el caso de móviles, con sistema operativo iOS compatibles con Safari y Android compatibles con Google Chrome

Tabla 5 - Criterios de aceptación para los RNF

6.4. Definición de MVP

En base a lo detallado en la sección [6.1.2. Priorización y reescritura](#), fue que definimos nuestro MVP. Para ser considerado como tal, debía incluir los cuatro primeros RF y todos los RNF, ya que dichas funcionalidades fueron validadas con los interesados y por lo tanto, consideradas suficientes para presentarlas como una versión mínima viable de nuestro producto y que el mismo tenga sentido.

Asimismo, se enfocan en los puntos que identificamos como necesidades del sector y su validación nos permitió conocer las opiniones de los usuarios y obtener información relevante acerca de su aporte de valor al rubro, antes de que el producto sea lanzado a producción.

El motivo por el cual consideramos que nuestro producto final puede ser llamado un MVP es que cumplimos con las condiciones que definimos, lo que permite que al momento de probarlo con usuarios, tengamos una plataforma con las funcionalidades clave y que representan la propuesta de valor del mismo.

Además, su elaboración tuvo mucha investigación y análisis por detrás, para asegurarnos de no solo incluir los requerimientos, sino también hacerlo de la mejor manera posible, teniendo en cuenta también el feedback que fuimos obteniendo de los interesados en las instancias de validación.

6.5. Conclusiones

En base a todo lo mencionado anteriormente, podemos concluir que el proceso de obtención de requerimientos es un proceso complejo, que requirió de mucho tiempo y esfuerzo de nuestra parte, ya que la correcta documentación es clave para el posterior desarrollo.

Al no contar con la experiencia profesional y ser la primera vez que se realiza este proceso en un proyecto de estas características, requerimos de tiempo para garantizar y validar que los RF y los RNF se alinearan con las necesidades del sector.

Asimismo, el hecho de tratarse de un proyecto emprendedor en el cual el público objetivo no es completamente fijo, el definir en etapas tempranas estos requerimientos resulta engorroso y tiene que poder estar diseñado para soportar cambios en la estructura de los mismos, a medida que avanza el desarrollo a la par de las validaciones.

7. Arquitectura y diseño

En este capítulo, describiremos la arquitectura y diseño de nuestra solución implementada, teniendo en cuenta los atributos de calidad y principales desafíos. También se incluye la descripción de tecnologías, servicios y estándares utilizados.

7.1. Descripción general de la Arquitectura

El objetivo de esta sección es introducir al lector sobre las consideraciones arquitectónicas que se tuvieron y la visión general de la arquitectura con sus elementos, la cual representa la solución obtenida como parte de este proyecto, dando paso a la justificación más detallada de las siguientes secciones.

7.1.1. Consideraciones arquitectónicas

Dado que este proyecto es de tipo emprendedor, el equipo tuvo la tarea elegir un correcto set de herramientas y servicios que permitieran construir una plataforma web de calidad, que pudiera adaptarse fácilmente a los cambios que surjan en un futuro. Esto debido a que uno de los objetivos de este proyecto es la realización de un MVP, el cual, en etapas posteriores a la entrega de este proyecto, probablemente tendrá modificaciones. Este set de herramientas surge como parte de una investigación tecnológica realizada antes y durante la construcción de nuestra plataforma web.

7.1.2. Visión general

Nuestra solución se constituye, por un lado, de un Frontend compuesto por una aplicación web desarrollada en React.js con TypeScript. Por otro lado, de un Backend compuesto por varios servicios de AWS, entre ellos, funciones con AWS Lambda y AWS Lambda@Edge desarrolladas con Node.js utilizando TypeScript, que nos permitieron configurar nuestra lógica de negocio. En la sección [7.4. Tecnologías](#), se explicará más en detalle la selección de estas herramientas de desarrollo.

Como primer desafío, tuvimos que escoger la plataforma para desplegar nuestra plataforma web. Para ello seleccionamos AWS, lo cual se debió a varios factores. En primer lugar, la experiencia previa del equipo con AWS y su amplia gama de servicios, entre ellos autorización y conversión de video, administración y búsqueda de datos y manejo de archivos, sumado a las características de integración que AWS ofrece,

fueron un factor clave para el desarrollo de nuestra plataforma. Además, AWS es conocido por su escalabilidad y rendimiento, lo que lo hace ideal para plataformas web de gran tráfico, ya que, al tratarse de un MVP, se espera que tenga una rápida adopción y crecimiento. Por este motivo, la calidad del servicio ofrecido no debe disminuir ante el aumento del tráfico y la carga de trabajo.

Otro factor importante al momento de escogerlo fue la facilidad de uso de Serverless Framework en el entorno de AWS. Esta herramienta nos permitió agregar y modificar fácilmente nuestros servicios de AWS, lo que, junto a una integración y despliegue continuo de estos, redujo la cantidad de acciones necesarias para enlazarlos y permitió al equipo enfocarse en el desarrollo de la plataforma web en sí. La rápida creación de nuevos ambientes tales como producción y desarrollo en simples pasos también contribuyó a lo anterior en gran medida.

Si bien Azure y Google Cloud también fueron considerados ya que ofrecen servicios y características similares a las utilizadas, la experiencia previa del equipo con AWS, su amplia gama de servicios y características de integración antes mencionadas, así como su escalabilidad y rendimiento, fueron factores determinantes en la elección de AWS como plataforma para desplegar nuestra plataforma web.

A su vez, el acceso a créditos para emprendedores nos permitió enfocarnos en el desarrollo de la plataforma sin incursionar en costos de infraestructura. Esto fue especialmente importante para nuestro proyecto, ya que los costos de infraestructura iniciales podrían haber sido una barrera importante para llevar adelante un emprendimiento de este tipo.

Sin embargo, AWS presentaba un problema; crear servicios y enlazarlos era una tarea compleja. Para solucionar esto, utilizamos Serverless Framework, ya que nos permite agregar servicios de AWS u otras plataformas fácilmente como parte de un ambiente específico, como se explicará más adelante en la sección [7.4 Tecnologías](#).

Teniendo en cuenta estos puntos, podemos pasar a detallar como interactúa nuestro Frontend con el Backend dentro de los distintos servicios (Figura 14). Es importante destacar que más adelante en este capítulo se explicarán detalles específicos sobre la arquitectura y este diagrama tiene el único fin de introducir al lector en la arquitectura general de la plataforma.

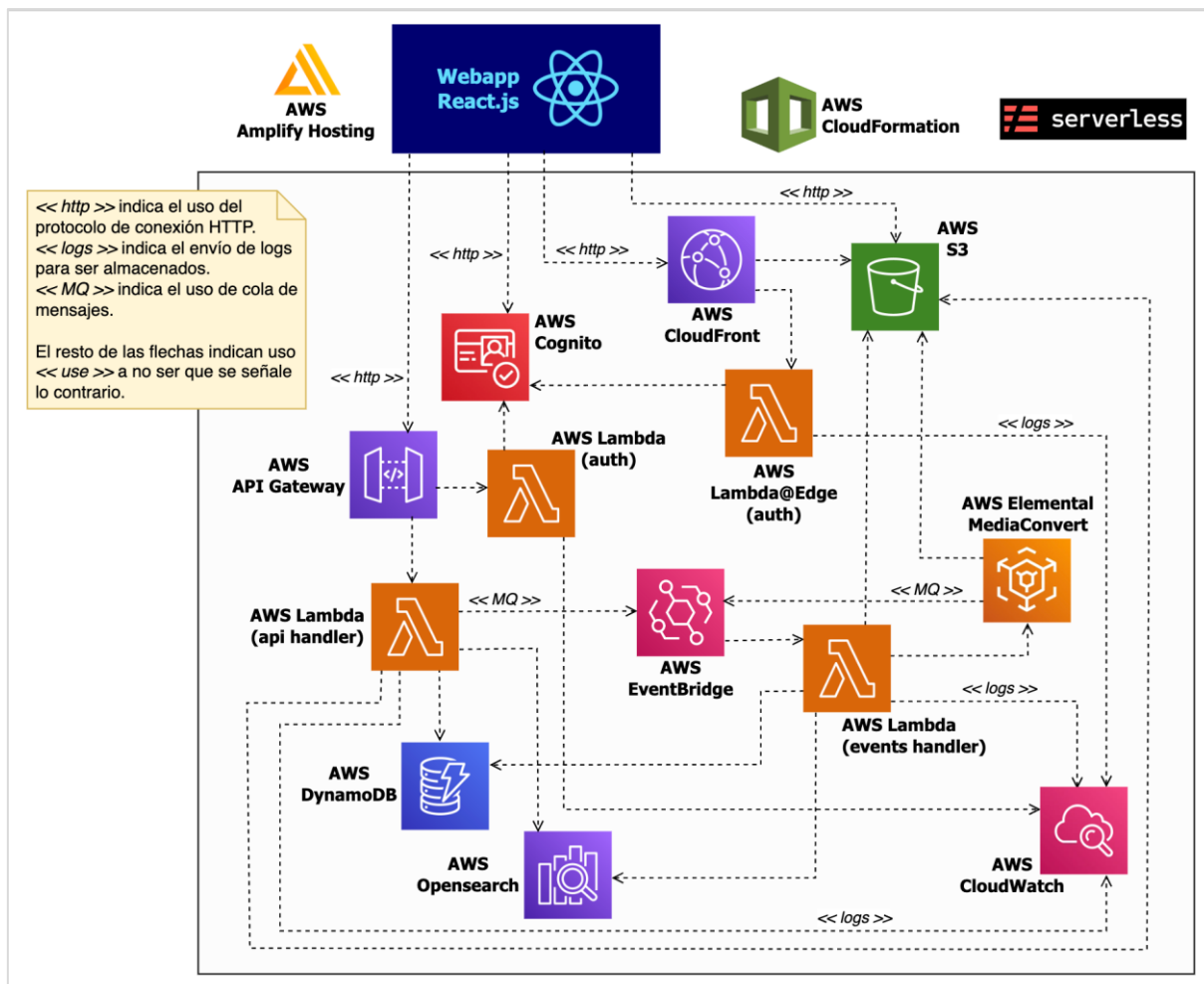


Figura 14 - Interacción del Frontend con el Backend

Como se puede apreciar a gran escala, el uso de los servicios de AWS nos permitió construir una plataforma más integrada, junto a la facilidad de despliegue que nos brinda Serverless Framework, utilizando AWS CloudFormation para el Backend y, AWS Amplify Hosting para el Frontend.

Nuestro Frontend está compuesto por la aplicación web o Webapp, desarrollada en React.js, que utiliza varios servicios para su correcto funcionamiento. En primer lugar, para la realización de acciones relacionadas con nuestros recursos, la Webapp accede a nuestra REST API implementada con AWS API Gateway, la cual está asociada a una instancia de AWS Lambda llamada (api handler). Sin embargo, antes de pasar por esta instancia, se valida la autorización correspondiente utilizando la función implementada en AWS Lambda llamada auth.

Además, para el manejo, validación y autenticación de usuarios, la Webapp utiliza AWS Cognito. Por último, para acceder a los archivos almacenados en AWS S3, se

utiliza AWS CloudFront. Antes de acceder a dichos archivos, se valida la autorización correspondiente utilizando una función implementada en AWS Lambda@Edge. Es importante destacar que, en el caso de que se deban subir archivos directamente a AWS S3, se utiliza una URL preasignada asociada a este servicio.

Las bases de datos son utilizadas por las funciones llamadas api handler y events handler, siendo estas instancias de AWS DynamoDB y de AWS OpenSearch.

Para el flujo de publicación de un curso nuevo o existente, se utiliza AWS EventBridge asociado a una instancia de AWS Lambda llamada events handler, encargada de manejar los eventos. Estos pueden ser emitidos por la función api handler, como es el caso del evento de publicación asíncrona de un curso, o por AWS MediaConvert al comenzar, fallar o terminar un trabajo de transcodificación de un curso. Más adelante se verá este flujo de publicación en la sección [7.6.2. Flujo de creación de un curso](#).

Como dato adicional, todos los eventos ocurridos en nuestras funciones son almacenados en AWS CloudWatch.

Para ver en más detalle la explicación de cada elemento ir a [13.7. Anexo 7: Catálogo de elementos del diagrama general](#). A su vez, si se desea visualizar la arquitectura inicial y final a través de un diagrama de componentes y conectores, ver [13.8. Anexo 8: Vistas de componentes y conectores](#).

7.2. Atributos de calidad

En esta sección se comentarán los principales atributos de calidad tomados en cuenta para nuestra plataforma y que tácticas utilizamos para cumplir con los mismos. Ambos, atributos y tácticas, se basan, en su mayoría, en los descritos en el libro Software Architecture in Practice [2]. En el caso de los atributos de Usabilidad y Portabilidad, estos serán justificados en la sección [10.4. Cumplimiento de atributos de calidad](#).

7.2.1. Deployabilidad

Este atributo se relaciona con la facilidad con la que se puede instalar y configurar el software en el entorno de producción, así como con la capacidad de actualización y despliegue de nuevas versiones de manera rápida y sin interrupciones.

- **Uso de Serverless Framework junto con CI/CD**

Pese a no ser una táctica concreta, Serverless Framework, junto con la integración y despliegue continuo utilizando GitHub Actions y AWS Amplify Hosting, nos permitieron desplegar y modificar servicios de forma rápida y en distintos ambientes, lo cual resulta clave para nuestro proyecto emprendedor. Estas tecnologías se ven en más detalle en la sección [7.4. Tecnologías](#).

7.2.2. Disponibilidad

Este atributo se refiere a la habilidad que tiene un sistema de estar disponible en el momento que un usuario desea utilizarlo, encontrándose esta en un estado que permita responder a la acción realizada por el usuario.

- **Manejo de excepciones**

Esta táctica busca el manejo de las excepciones una vez que estas ocurren, evitando que el sistema se detenga. En nuestro caso, aplicamos esta táctica en nuestras funciones, tanto en api handler como en events handler, donde se implementa un manejador de errores transversal a todas las capas que se verán en la sección [7.6.1. Vista de módulos](#), en el punto correspondiente a Vista de módulos Layers. A su vez, se manejaron excepciones en el Frontend, para controlar las llamadas a nuestra API, las llamadas a AWS Cognito, los errores del reproductor de video y la subida de archivos a AWS S3.

- **Prevención de excepciones**

Esta táctica tiene como objetivo el intentar evitar que ocurra una excepción en nuestro sistema. Para ello se brinda en nuestra aplicación web, ayudas para prevenir dichas excepciones, tales como el uso de la guía en la visualización de videos de realidad virtual y la validación de los datos ingresados en los campos de texto de los formularios.

- **Disponibilidad de servidores**

Esta no es una táctica descrita en el libro, pero si está relacionada directamente con la disponibilidad y es el uso de servicios con alta disponibilidad.

A continuación, en la Tabla 6, mencionamos los servicios de AWS que usamos y su respectiva disponibilidad (SLA). [9]

Servicio	SLA
API Gateway	99,95
CloudFront	99,9
Lambda	99,95
DynamoDB	99,99
S3	99,99
Cognito	99,9
EventBridge	99,9
MediaConvert	99,99
CloudWatch	99,99
OpenSearch	99,95

Tabla 6 - Servicios de AWS utilizados y su respectiva SLA

Multiplicando sus valores para obtener la disponibilidad en el peor caso que se usen todos a la vez, obtenemos una disponibilidad de 99,55%, por lo cual podemos afirmar el cumplimiento del RNF1 – El sistema debe tener una disponibilidad del 95% de las veces en que un usuario intente accederlo.

7.2.3. Eficiencia

Este atributo se refiere al tiempo y la capacidad del sistema de software para cumplir con los requisitos de tiempo.

- **Introducir concurrencia**

En nuestro caso, mediante el uso de AWS CloudFront manejamos varias solicitudes a la vez para obtener archivos, como se menciona en la sección [7.3.4. Cumplimiento de Twelve Factor App](#). Con esto podemos afirmar el cumplimiento de la concurrencia y por lo tanto, del RNF2 – El sistema deberá poder soportar 100 usuarios visualizando videos de forma simultánea.

- **Utilización de servicios a demanda**

A pesar de no ser una táctica en particular, la estrategia utilizada para poder cumplir con el procesamiento de videos en distintas calidades fue recurrir al mismo mecanismo que utilizan varios sitios de reproducción de video: reproducir a demanda.

Esto significa que en lugar de cargar todo el video y luego reproducirlo, este se va particionando y cargando por partes, disminuyendo así el tiempo de espera que tiene el usuario al visualizar un video y cargando de a poco los siguientes fragmentos a la vez que mira los anteriores. El problema que presenta esto es que se tiene que particionar el video en un formato que permita ser unido fácilmente en tiempo real, equiparando la experiencia de visualización a la de ver un video descargado de antemano.

En nuestro caso, para lograr esto utilizamos AWS MediaConvert y creamos un template con la configuración necesaria para transformar videos comunes de un archivo (como puede ser .mp4 o .avi) al formato .m3u8, recomendado y desarrollado por Apple para la transmisión de video a sus dispositivos. Dicho template particiona los videos y a su vez los pone en distintas calidades, por lo que, si la conexión a internet del usuario cambia o si el usuario quiere seleccionar una calidad en particular, el reproductor tiene la opción de cargar la siguiente partición de video en una calidad más alta o baja, dependiendo el caso. La partición que hace AWS MediaConvert transforma un video .mp4 u otro formato a un archivo .m3u8 que especifica el nombre de las siguientes partes a cargar del video, de principio a fin.

La librería utilizada desde nuestra aplicación web para lograr esto es video.js, junto con otras extensiones necesarias para este tipo de reproducción de video, que reconocía el primer archivo y cargaba automáticamente los siguientes fragmentos.

Sin embargo, otro problema que se presentó fue que el template debía ser configurado de distinta forma para los videos 360. Por ello, creamos un template diferente para estos tipos de video.

El proceso llevado a cabo para la conversión de estos videos se explica en el diagrama de flujo de la sección [7.6.2. Flujo de publicación de un nuevo curso](#).

- **Mantener múltiples copias de datos (Caching)**

Para consumir dichos videos, además de utilizar la librería video.js, usamos AWS CloudFront, un CDN que nos permite mejorar la velocidad y la disponibilidad de la entrega de contenido web a los usuarios finales. Esto lo logra almacenando en caché contenido en servidores en múltiples ubicaciones geográficas, acelerando la entrega

de dicho contenido y reduciendo la latencia para los usuarios finales, mejorando así la experiencia de usuario. Asimismo, está alineado con los atributos de calidad de Disponibilidad y Seguridad de la plataforma web, como se explica en las secciones [7.2.2. Disponibilidad](#) y [7.2.5. Seguridad](#) respectivamente.

Es importante destacar que la utilización de AWS Lambda@Edge nos permite ejecutar código en los puntos de presencia de la red global de CDN de AWS, a diferencia de AWS Lambda que tiene regiones fijas. Por esto, el uso de este servicio para autorizar a los usuarios como se menciona en la sección [7.2.5. Seguridad](#), no afecta el funcionamiento de AWS CloudFront en gran medida.

A su vez, otra aplicación a esta táctica fue el uso del motor de búsqueda AWS OpenSearch, utilizado como estrategia para corregir el trade-off que presenta AWS DynamoDB como base de datos. Esto debido a que las búsquedas no son eficientes y el uso de índices solo permite mejorarlas un poco. Además, el problema surge cuando queremos priorizarlas, volviéndose muy difícil de desarrollar y pudiendo tener una demora excesiva, provocando largas esperas al usuario. AWS OpenSearch soluciona esto, permitiendo realizar búsquedas en base a criterios en forma eficiente, como se explica en detalle en la sección [7.5. Gestión de datos](#). Este contiene un sub-set de datos, en nuestro caso, información relevante a un curso, y es actualizado cada vez que un elemento de dicho sub-set del curso cambia.

Con todas las tácticas anteriormente mencionadas, podemos afirmar el cumplimiento del RNF3 – El sistema deberá manejar de forma eficiente el procesamiento de vídeo.

7.2.4. Modificabilidad

Este atributo tiene como fin poder implementar cambios en el sistema, reduciendo el impacto que estos generan al menor costo posible. Pese a que las tácticas utilizadas explicadas a continuación para cumplir con este atributo no están asociadas a un RNF en particular, se aplicaron con el fin de construir una plataforma web de calidad.

- **Encapsulamiento**

Esta táctica busca introducir interfaces explícitas para la utilización de un módulo en particular. La misma fue aplicada junto con la vista de módulos Layers mencionada en

la sección [7.6.1. Vistas de módulos](#), ya que define interfaces explícitas para la utilización de estos módulos en particular.

- **Uso de intermediarios**

La idea de esta táctica es romper dependencias entre elementos mediante la creación de un elemento extra que sirva para conectar ambas partes. En nuestro caso utilizamos Redux, como se menciona en la sección [7.3.2. Manejo de datos en el Frontend](#), separando la lógica de obtención y modificación de datos (productor) del componente que muestra esos datos o realiza esas acciones (consumidor).

- **Abstracción de servicios comunes**

Esta táctica busca generalizar o abstraer servicios comunes. En nuestro caso, hicimos uso de una carpeta *common* dentro del repositorio de servicios, descrito en la sección [7.6.1. Vista de módulos](#), dado que reutilizamos diferentes implementaciones en varias funciones.

- **División de Módulos**

Esta táctica busca separar módulos complejos, debido a que el costo de modificarlos se vuelve muy costoso. En nuestro caso, para las funciones se utilizó una división en capas base para dividir responsabilidades como se puede ver en la sección [7.6.1. Vista de módulos](#), en el punto correspondiente a Vista de módulos Layers, siendo una mala práctica, común en la industria, encontrarse con funciones que contienen toda la implementación en un solo archivo. En el caso de nuestra aplicación web, hicimos una separación de componentes y contenedores como se puede ver en la sección [7.6.1. Vista de módulos](#), en el punto correspondiente al Frontend.

- **Uso de un único lenguaje en el Frontend y Backend**

Pese a no ser una táctica en sí, el uso de un único lenguaje como se verá más adelante en la sección [7.4. Tecnologías](#), reduce la complejidad del proyecto en términos de la cantidad de tecnologías y herramientas utilizadas. Esto hace que el proyecto sea más fácil de entender y de mantener.

- **Uso de bases de datos no relacionales.**

Pese a no ser una táctica en sí, el uso de bases de datos no relacionales, en nuestro caso AWS OpenSearch y AWS DynamoDB, como se explica en detalle en la sección [7.5. Gestión de datos](#), nos permiten hacer cambios en las estructuras de los datos de una manera muy simple, sin tener que hacer migraciones complejas como es el caso de las bases de datos relacionales. De esta forma, es más rápido adaptarse al cambio que puede surgir en el modelado de datos, siendo esto algo recurrente en proyectos de tipo emprendedor.

7.2.5. Seguridad

La seguridad se refiere a la capacidad que tiene un sistema de permitirle a los usuarios legítimos consumir datos y servicios y que los usuarios no legítimos no puedan realizar dichas acciones.

- **Autorizar autores**

Esta táctica busca verificar que un usuario tenga los permisos necesarios para acceder o modificar un recurso de nuestro sistema. En nuestro caso utilizamos las funciones AWS Lambda y AWS Lambda@Edge para validar el acceso a nuestra API y a nuestro bucket de AWS S3 mediante la utilización de JSON web tokens (JWT), validando el origen y el producto de la request (como se verá en la sección [7.3.5. Arquitectura Multi-Tenant](#)). Desde nuestra aplicación web se crearon restricciones con respecto a la navegación para los distintos casos en donde el usuario inicia sesión, inicia sesión como administrador o no inicia sesión.

- **Limitar exposición y acceso**

Esta táctica busca limitar el acceso a nuestro sistema o componente. Para ello, configuramos el bucket de AWS S3 como privado y creamos permisos de AWS IAM dentro de Serverless para autorizar el acceso a determinados servicios dependiendo del origen. Sin embargo, sigue siendo necesario subir archivos, por lo cual, AWS S3 genera una URL preasignada segura que permite únicamente subir el tipo de archivo especificado antes de generar dicha URL.

- **Encriptar la información**

Esta táctica tiene como fin que la información se encuentre protegida de accesos no autorizados. Para ello, mediante AWS API Gateway y AWS Amplify Hosting, utilizamos HTTPS para cifrar la comunicación entre cliente y servidor.

- **Actores Autenticados**

Esta táctica busca garantizar que los actores (usuarios) sean quien dice ser. En nuestro caso, implementamos un login utilizando AWS Cognito, el cual nos provee una implementación de manejo de registro e inicio de sesión, entre otras funciones.

- **Identificar actores**

Esta táctica busca tener un mecanismo para identificar el origen de las entradas externas al sistema. Para ello, utilizamos el correo electrónico como valor único y cuando se registra, este pasa a ser el identificador del usuario dentro del sistema, junto con el producto que está siendo utilizado.

Debido a las tácticas anteriormente mencionadas, podemos afirmar que cumplimos con el RNF4 – Se deberá proveer mecanismos de seguridad que garanticen la integridad de los archivos que contiene un curso.

7.3. Estándares

En esta sección detallaremos algunos estándares utilizados, que nos permitieron construir una plataforma web de calidad.

7.3.1. Sistema de diseño

Para el desarrollo de estilos de plataformas web, el tipo de estilización utilizado por defecto es CSS (hojas de estilo en cascada). Estas nos permiten reutilizar varios estilos, pero algunos de los problemas que presentan son el evitar la repetición de los mismos y el lograr que brinden una buena experiencia de usuario. [10] Para solucionarlo, existen los sistemas de diseño, los cuales se pueden definir como “... un conjunto de directrices basadas en patrones y buenas prácticas del desarrollo y diseño, cuidadosamente organizadas y documentadas, que buscan gobernar el desarrollo de la UI de un producto digital...”. [11]

En nuestro caso utilizaremos el sistema de diseño Material Design, el cual según su propia definición “...se basa en las mejores prácticas en los campos del diseño web y tradicional, y se basa en la investigación de la experiencia del usuario y la ciencia cognitiva. Las pautas de diseño que se desarrollaron a partir de los hallazgos de la investigación están destinadas a aplicarse universalmente en todas las plataformas y dispositivos.” [12]

Al utilizar React.js para el desarrollo de nuestra plataforma, de forma de poder acoplarnos a este sistema de diseño, utilizamos la librería material-ui en su versión 4, la cual, como se demostró en el trabajo final de carrera “Tecnologías de software para interfaz de usuario adaptativa y multiplataforma”, se acopla a los principios y lineamientos seguidos por Material Design, permitiendo un correcto manejo y versionado de los estilos del sistema, además de brindar componentes de uso común, como pueden ser Box, Grid y Button. [13] [14]

- **Paleta de colores**

Material Design ofrece una herramienta como guía para seleccionar una paleta de colores a utilizar en un sistema, con el fin de cumplir con los lineamientos que establece el mismo. De esta forma, se define un color primario y otro secundario con sus variantes de tonalidades (Figura 15). [15]

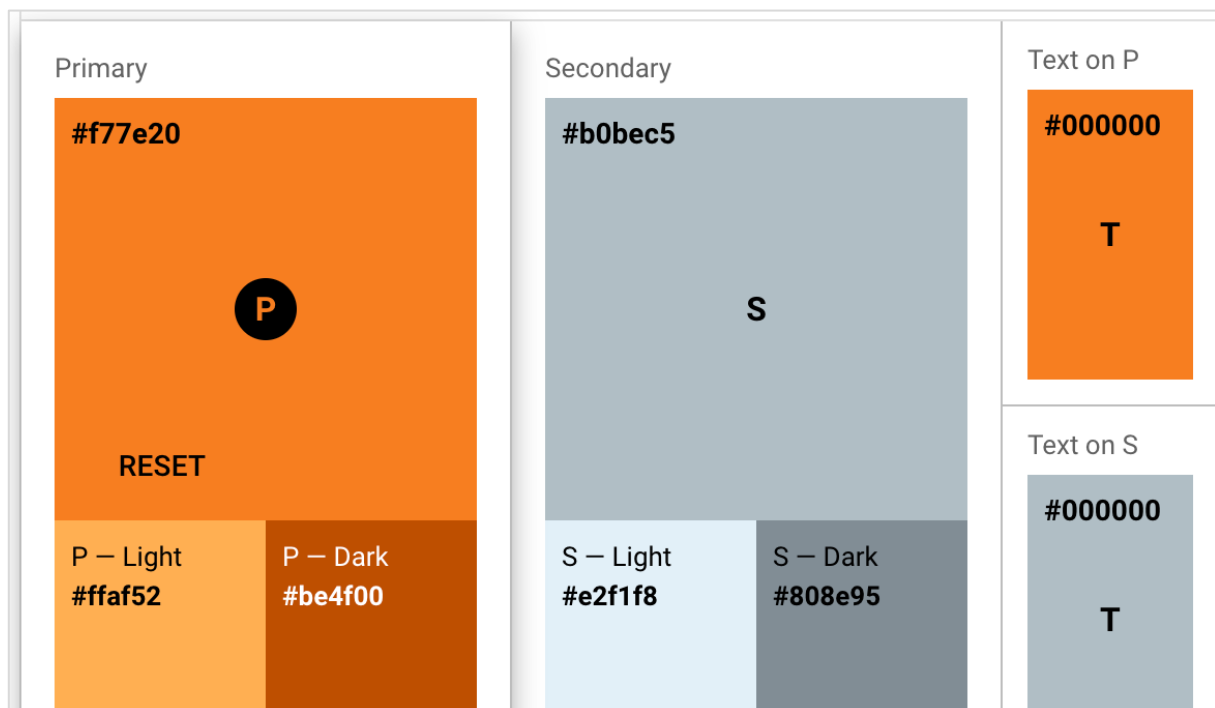


Figura 15 - Paleta de colores escogida con la herramienta de color de Material Design

Hacemos uso de esta paleta de colores en la mayoría de los elementos de nuestra plataforma. Sin embargo, existen excepciones en donde utilizamos distintos colores para representar errores o mensajes de éxito. Estas excepciones se encuentran contempladas por la librería material-ui.

Es importante destacar que esta paleta de colores fue seleccionada en base a una selección previa de colores utilizados en la industria y una posterior aprobación de parte de los interesados.

7.3.2. Manejo de datos en el Frontend

En una aplicación web construida con React.js y basada en componentes, el pasaje de datos entre ellos puede ser algo tedioso y muy complejo. Para solucionar esto, utilizamos una aplicación de los contextos de las más populares en este tipo de aplicaciones: Redux, el cual permite el manejo del estado de nuestra aplicación de una forma fácil y consistente. [16]

Redux se basa en la arquitectura Flux provista por Facebook para la creación de aplicaciones cliente-servidor, por lo que, el seguimiento de estos estándares nos garantiza el uso de una librería utilizada en la industria, en constante actualización y testeada correctamente. [17] Para ver la estructura de datos almacenada en dicho contexto ver [13.8. Anexo 8: Estructura de datos en Redux](#).

7.3.3. REST API

Para poder explicar que es una REST API, primero es importante definir brevemente algunos conceptos:

- REpresentational State Transfer o REST, es un estilo arquitectónico que describe interfaces uniformes entre componentes de software desacoplados. Inicialmente fue definido por Roy Fielding en su tesis de doctorado. [18]
- APIs son un mecanismo de comunicación entre componentes de software. [19]

Definidas ambas partes, podemos decir que, para que un servicio sea considerado una REST API, tiene que ofrecer un mecanismo de comunicación entre componentes de software que cumpla con el estilo arquitectónico REST. Para esto, es necesario un

protocolo de comunicación como lo es HTTP. En nuestro caso, utilizamos el servicio AWS API Gateway para recibir estas consultas y luego delegarlas en formato de eventos a nuestra función en AWS Lambda llamada api handler, como se menciona en la sección [7.1.2. Visión general](#). La misma, al cumplir con el estilo arquitectónico REST, nos permite exponer nuestros recursos. Con esto, podemos realizar acciones desde nuestra aplicación de manera asincrónica y síncrona sobre nuestros datos, de una manera simple y estructurada mediante el uso del protocolo HTTP, dividiendo nuestros datos en recursos. También, nos permite agregar nuevas acciones a estos últimos de forma sencilla y organizada. Esto resulta muy relevante para nuestra plataforma, ya que el acceso a datos de una forma estandarizada nos permite separar y abstraer responsabilidades, fomentando la mantenibilidad y extensibilidad de nuestra plataforma, atributos clave para un proyecto de tipo emprendedor.

En cuanto al diseño de los recursos, seguimos los lineamientos expuestos en el libro “REST API Design Rulebook”. [19] Para ver el diseño de recursos y endpoints de nuestra API y su relación con los requerimientos funcionales ir a [13.10. Anexo 10: Endpoints](#).

7.3.4. Cumplimiento de Twelve Factor App

Twelve Factor App es un conjunto de buenas prácticas para el desarrollo y arquitectura de servicios basados en la nube. [20] En nuestro caso, el objetivo de estos principios se alinea con nuestro proyecto de tipo emprendedor, donde es necesario poder escalar fácilmente y estar preparado para poder ser distribuido, construyendo así una plataforma web de calidad. Los siguientes puntos detallarán si se cumplió con estos principios.

- **Código Base**

Este principio se cumple ya que se realizan los despliegues automáticamente desde los dos repositorios que tenemos en GitHub, utilizando Git como mecanismo de control de versiones.

- **Dependencias**

Este principio se cumple ya que todas las dependencias son explícitas y existe un sistema (Node.js) que se encarga de descargarlas e instalarlas, tanto para las

funciones como para la aplicación web, ya que ambos se basan en dicho sistema y su manejador de paquetes, Node Package Manager (NPM).

- **Configuraciones**

Podemos afirmar el cumplimiento de este principio puesto que las variables de entorno, tales como el ambiente y credenciales de AWS, son configuradas mediante las interfaces disponibles en AWS. En el caso de la aplicación web, AWS Amplify Hosting permite cambiar y configurar variables de entorno dependiendo del ambiente. En el caso de las funciones, se leen de AWS Parameter Store, por lo cual pueden modificarse.

- **Backing Services**

La conexión a las bases de datos se realiza mediante el uso de variables de entorno, dependiendo del ambiente, por lo cual afirmamos que se cumple este principio.

- **Construir, Distribuir y Ejecutar**

Estas etapas se realizan de forma automática al mergear a las ramas “master” (ambiente de producción) y “develop” (ambiente de desarrollo). En el caso de la aplicación web, AWS provee un mecanismo que es compatible con la integración y despliegue continuo de forma automática, dependiendo el ambiente. En el caso de las funciones, utilizamos GitHub Actions para hacer la integración y despliegue continuo también dependiendo el ambiente. Por estos motivos, podemos afirmar que este principio se cumple.

- **Procesos sin estados**

Al tener una API que cumple con REST y la información relevante almacenada en las bases de datos, podríamos decir que se cumple, pero, las funciones AWS Lambda pueden guardar alguna información entre eventos, por lo que no se cumple con este principio enteramente. [18]

- **Asignación de Puertos**

AWS se encarga de la asignación interna de puertos y nuestra aplicación es compatible con esto, pero las lambdas no exponen redes directas, por lo que este principio no se cumple. [21]

- **Concurrencia**

Las funciones AWS Lambda fueron creadas para manejar la concurrencia masiva, escalando automáticamente a pesar de ser un mecanismo distinto al tradicional con máquinas virtuales o host. [21] Las bases de datos también soportan concurrencia y la aplicación web vive principalmente en el navegador (cliente) utilizado por el usuario. En el caso de AWS CloudFront, como dice su documentación, soporta hasta 250.000 solicitudes por segundo [22]. Por estas razones, podemos afirmar que se cumple parcialmente este principio.

- **Disponibilidad**

Las lambdas no se apagan, el inicio de los servicios de AWS, en caso de no estar en uso, se podría considerar como rápido y el apagado es automático en caso de que no se utilice. Por ello, podemos afirmar el cumplimiento parcial de dicho principio. [21]

- **Igualdad entre desarrollo y producción**

Se realizaron despliegues automáticos y continuos a los ambientes de producción (rama “master”) y desarrollo (rama “develop”). A su vez, se utilizan distintos servicios, pero con la misma configuración en cuanto a recursos. Por estas razones, podemos afirmar el cumplimiento de este principio.

- **Logs**

Se guardan los logs de los servicios de AWS utilizando AWS CloudWatch, separándolos por fecha, ambiente y tipo de log, pudiendo ser estos “Error”, “Debug” o “Info” (Informativo) poniendo luego el mensaje de error y finalmente el objeto o error sucedido. Por esto, podemos afirmar el cumplimiento de este principio.

- **Procesos de administración**

Este principio no se cumple debido a que no tenemos procesos de administración y no se aplican a las funciones de tipo AWS Lambda. [21]

7.3.5. Arquitectura Multi-Tenant

Según su definición, Multi-tenancy se refiere a una arquitectura en donde el software puede ser utilizado por varios clientes, cada uno con sus datos separados del resto de forma lógica o física. [23] En nuestro caso, al no tener un cliente fijo, puede ser necesario que varias empresas utilicen nuestro sistema por separado, por lo cual este tipo de arquitectura es clave para nuestra plataforma.

Debido a que nuestra prioridad es minimizar el costo, tanto de desarrollo como de servicios al tratarse de un MVP, optamos por la implementación de Multi-Tenant App con Multi-Tenant DB, utilizando una separación totalmente lógica. Esto implica que se utilicen los mismos servicios, a excepción de AWS Cognito, que tiene su set de usuarios distinto y la aplicación web que tiene instancias separadas debido al dominio, siendo esta la forma interna que usan nuestras funciones de autorización para enlazar un origen a determinado producto. Para lograr esta implementación se agregó a la clave primaria de cada tabla el producto al que corresponde. A modo de ejemplo, si el correo electrónico es "prueba@prueba.com", el "userId" en la tabla "Users" para el producto "X" será "X-prueba@prueba.com". Además, cada documento contiene la referencia al producto como se puede ver en [13.11. Anexo 11: Tablas y Colecciones](#).

Debido a que una de las grandes propuestas de valor de nuestra plataforma son las distintas formas de visualizar videos, es posible, si el cliente lo desea, cambiar el template de conversión de videos en AWS MediaConvert, pudiendo configurar una versión para videos 360 y otra para los videos normales o con varios ángulos de cámara. Para ver una guía de cómo crear un nuevo producto ver [13.12. Anexo 12: Crear un nuevo producto](#).

7.4. Tecnologías

Debido a que el proyecto está compuesto por dos integrantes y ambos debíamos participar en las tareas relacionadas al Frontend y Backend, se buscó que estos utilizaran tecnologías similares.

Las tecnologías que cumplían con esto son Node.js, para el desarrollo de las funciones con AWS Lambda y AWS Lambda@Edge, y React.js, para el desarrollo de aplicaciones web, el cual a bajo nivel también utiliza Node.js. Esto debido a que ambas partes usan el mismo lenguaje y son compatibles con librerías que nos permiten transmitir video, y en el caso de React.js, conectarnos a los servicios de AWS y seguir un sistema de diseño como se vio en la sección [7.3.1. Sistema de diseño](#).

Asimismo, el equipo tenía experiencia en cuanto al uso de Node.js para la implementación de servicios y de React.js para la creación de aplicaciones web. Gracias a esto, pudimos enfocarnos en el desarrollo de la plataforma en sí, sin tener que invertir tiempo en aprender estas tecnologías de cero.

La ventaja que representa utilizar un mismo lenguaje como JavaScript (lenguaje que utiliza Node.js) para desarrollar en toda la plataforma, es que ayuda a cometer menos errores, ya que no se deben tener en cuenta distintos estándares de los lenguajes nativos. Sin embargo, presentaba un problema; JavaScript es no tipado, por lo que teníamos que hacer múltiples validaciones de los tipos reiteradas veces y, de todas formas, esto no garantizaba que no hubiera errores de tipado, los cuales podían provocar que el sistema se detenga. Por ello, la solución que encontramos fue la utilización de TypeScript, que no es más que JavaScript pero con la sintaxis de tipado, para todas las funciones y la aplicación web. [24]

Trabajar con un lenguaje tipado tuvo varios beneficios, entre ellos, no tener que validar constantemente el tipo de los datos, permitiéndonos escribir un código más claro y fácil de mantener. A su vez, el uso de interfaces y clases abstractas nos permite usar el Polimorfismo de una forma más definida y alineada a otros lenguajes orientados a objetos que hemos utilizado a lo largo de toda la carrera.

Es importante destacar que se usaron varias librerías externas que nos permitieron la comunicación simple con servicios de AWS, seguimiento del sistema de diseño

seleccionado, entre otros (para ver más detalle ir a la sección [7.3. Estándares](#)). A su vez, el detalle de las librerías más relevantes utilizadas puede encontrarse en [13.19. Anexo 19: Librerías externas](#).

Otro punto importante es el uso de Serverless Framework para detallar los servicios de AWS a utilizar y desplegarlos para distintos ambientes. Esto facilita la integración entre servicios dentro de un ambiente y nos permite agregar nuevos servicios o funciones con solo añadir líneas de código y subirlas a nuestro repositorio. Luego, en el siguiente despliegue, estos servicios serán creados y configurados para ese ambiente en particular, pudiendo ya ser utilizado. Este tipo de herramientas son especialmente útiles en este tipo de proyectos emprendedores con una arquitectura cambiante, ya que permiten hacer cambios rápidamente y con menos errores, reduciendo el tiempo necesario para dichos cambios de días a horas. Es importante destacar que para el despliegue continuo se utilizaron GitHub Actions en el Backend y se configuró AWS Amplify Hosting para el Frontend.

7.5. Gestión de datos

En la siguiente sección justificaremos, teniendo en cuenta las características del proyecto, la selección de bases de datos y el contenedor de archivos. Para ver sus tablas y colecciones ver [13.11. Anexo 11: Tablas y Colecciones](#).

- **DynamoDB**

AWS DynamoDB es una base de datos NoSQL de tipo clave-valor, la cual nos permite la ausencia de un esquema de datos fijo, pudiendo definir sus atributos en tiempo de ejecución, sin necesidad de hacer tediosas migraciones debido a un nuevo campo en el esquema, ya que se mantendrán los valores anteriores. Además, puede escalar horizontalmente fácil y automáticamente al agregar más nodos o clústeres para mejorar el rendimiento, la velocidad y el almacenamiento.

La mayor desventaja que presenta este tipo de base de datos es cuando se realizan consultas que involucran distintas tablas, como puede ser un JOIN, pero, dadas las características de nuestra plataforma, este tipo de consultas no serán necesarias. A su vez, para que las búsquedas dentro de una tabla sean más eficientes, AWS DynamoDB ofrece Índices para realizar dichas búsquedas en base a otros atributos.

Por estas razones, es que esta base de datos se acopla perfectamente a nuestra plataforma y a las características de un emprendimiento, donde escalar fácilmente es necesario y los cambios en los esquemas son recurrentes.

- **OpenSearch**

AWS OpenSearch es un motor de búsqueda y análisis de código abierto, el cual nos permite implementar clústeres de OpenSearch en la nube, pudiendo operar y escalar estos de manera sencilla [25].

En nuestra plataforma, al guardar los datos en AWS DynamoDB, las búsquedas por varios atributos a la vez que pueden no ser parte ni de la clave ni de índices son muy costosas, aún sin tener en cuenta la necesidad de priorizar los documentos que tengan más coincidencias de un atributo, frente a otro que coincida con el criterio buscado. OpenSearch resulta ser una solución muy útil, ya que nos permite guardar los datos que se necesiten analizar en un futuro, pudiendo hacer búsquedas por un criterio, priorizando las coincidencias en ciertos atributos. En nuestro caso, utilizamos OpenSearch para la búsqueda de un curso en base a su nombre, docente y descripción, en ese orden de priorización. También guardamos los nuevos cursos al crearlos y publicarlos.

- **AWS S3**

Necesitábamos un contenedor en la nube para poder almacenar los documentos y videos (convertidos y sin convertir) de un curso, los ejercicios entregados y los currículums enviados por los usuarios interesados en crear cursos. Para ello utilizamos AWS S3, puesto que es un excelente organizador de archivos y se encuentra dentro de los servicios ofrecidos por AWS, garantizando una excelente integración a estos. En nuestro caso, lo utilizamos con AWS CloudFront para mostrar el contenido de manera segura y con AWS MediaConvert para obtener los videos a convertir y guardar el resultado de dicha conversión. Asimismo, lo integramos con nuestras funciones que utilizan AWS Lambda para realizar operaciones como puede ser el borrado de videos.

7.6. Detalles de la arquitectura

En esta sección, se detallarán vistas de módulos de nuestra plataforma y un diagrama de flujo. La primera es una vista de módulos con el fin de visualizar las unidades de implementación que proveen un conjunto coherente de responsabilidades, en este caso, las funciones utilizando AWS Lambda y AWS Lambda@Edge, como se menciona en la sección [7.1.2. Visión General](#) y, el Frontend de nuestro sitio. Luego, detallaremos el flujo de publicación de un curso. [26]

Debido a la utilización de Serverless Framework para el despliegue de todas las partes de nuestra plataforma, no hay una vista de asignación, ya que un diagrama de despliegue no tendría sentido dado que el Framework es el encargado de este procedimiento. Es importante destacar que, a la hora de realizar los diagramas, buscamos seguir los lineamientos y elementos establecidos por el lenguaje unificado de modelado o por sus siglas UML, en su versión 2.0. [27]

7.6.1. Vista de módulos

En esta sección se visualizarán las vistas lógicas de nuestra plataforma web. [26]

- **Backend**

En el siguiente diagrama veremos todas las unidades lógicas que creamos para nuestro Backend sin contar los servicios que configuramos (Figura 16).

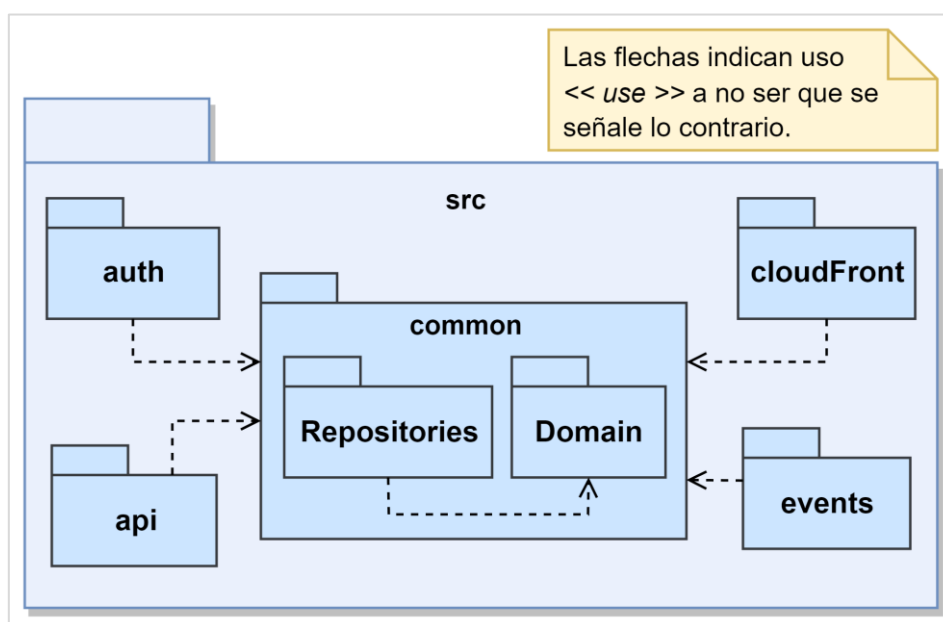


Figura 16 – Descomposición del módulo de Backend

A continuación, describiremos los elementos utilizados:

- **api**: Función que contiene la implementación de una REST API con las rutas, controladores y servicios necesarios para el funcionamiento del sistema.
- **auth**: Comprende la función que permite o niega la entrada a la API, por lo tanto, brinda un servicio de autorización a la misma.
- **cloudFront**: Engloba la función que permite o niega el acceso a los archivos dentro de la instancia de AWS S3 por medio de AWS CloudFront, brindando un servicio de autorización a la misma.
- **common**: Contiene información común a todas las funciones con el fin poder ser reutilizada, como puede ser clases del dominio, implementaciones de repositorios y constantes.
- **Domain**: Incluye las clases del dominio de nuestra plataforma y sus validaciones correspondientes.
- **events**: Contiene la función que maneja los distintos tipos de eventos que son emitidos mediante AWS EventBridge.
- **Repositories**: Posee la implementación concreta del acceso a datos de nuestro sistema.
- **src**: Incluye todas las funciones provistas por el sistema y el módulo common.

La decisión de ubicar todas las funciones en el mismo repositorio se debe a que comparten varias implementaciones. Esta forma favorece la mantenibilidad, ya que se evita que haya implementaciones duplicadas y cualquier cambio en estas forzarán a actualizar todas sus dependencias. Debido a como los servicios de AWS utilizan cada función, es necesario dividirlos como se ilustró en la Figura 16, ya que por más que se pueden segregar fácilmente en varias funciones, por ejemplo, en el caso de api handler, crear varias funciones por endpoint o, en el caso de events handler, crear varias funciones por evento, esta complejidad no aporta valor al MVP y complejiza la solución sin sentido. Además, al tratarse de un MVP, esta puede no ser la estructura final y podría sufrir cambios en las etapas posteriores a la finalización del proyecto.

▪ Vista de módulos Layers

El modelo de capas que se verá a continuación se utiliza tanto para api handler como para events handler (Figura 17).

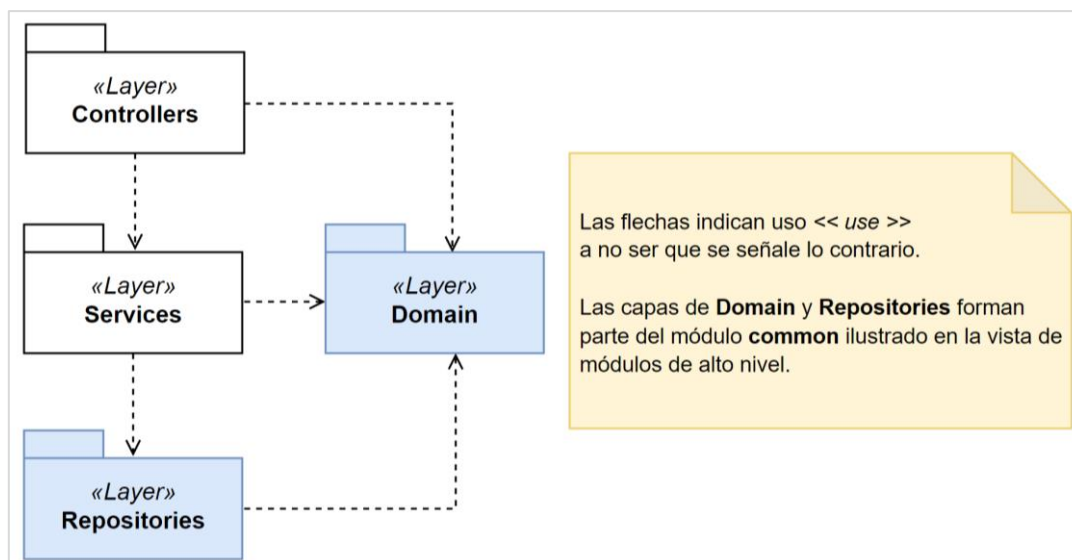


Figura 17 - Diagrama de capas

Este diseño cumple con el patrón arquitectónico Layer, favoreciendo la modificabilidad del sistema, ya que, si hubiera cambios a futuro, por ejemplo, en la base de datos utilizada, esto no afectaría la lógica de negocio que se encuentra en la capa de servicio. [2]

Es importante destacar que los repositorios y el dominio están dentro del módulo common, pero de todas formas esto no afecta el cumplimiento y los usos de la jerarquía de capas.

Para lograr que se respeten los accesos, se definen interfaces entre Controllers y Servicios y, entre Servicios y Repositorios, cumpliendo de esta forma con el principio SOLID de inversión de dependencias, utilizando inyección de dependencias. Para ver en detalle la aplicación de dichos principios ir a [13.13. Anexo 13: Diagrama de paquete de capas](#).

A continuación, listaremos los elementos utilizados:

- **Controllers:** Es responsable de gestionar y exponer la funcionalidad de dicho módulo. Cuando le llegan peticiones es quien las procesa y las deriva hacia la capa de servicio.

- **Domain:** Es responsable por definir la estructura de los registros, los campos que contendrá y el tipo de esos campos.
- **Services:** Es el responsable por gestionar y llevar adelante la lógica del negocio.
- **Repositories:** Es el responsable por gestionar la conexión con la base de datos, cargar los modelos, así como guardar, actualizar, recuperar o eliminar registros de la base de datos.

- **Frontend**

En el siguiente diagrama veremos cómo se descompone el módulo Webapp correspondiente a nuestra aplicación web (Figura 18).

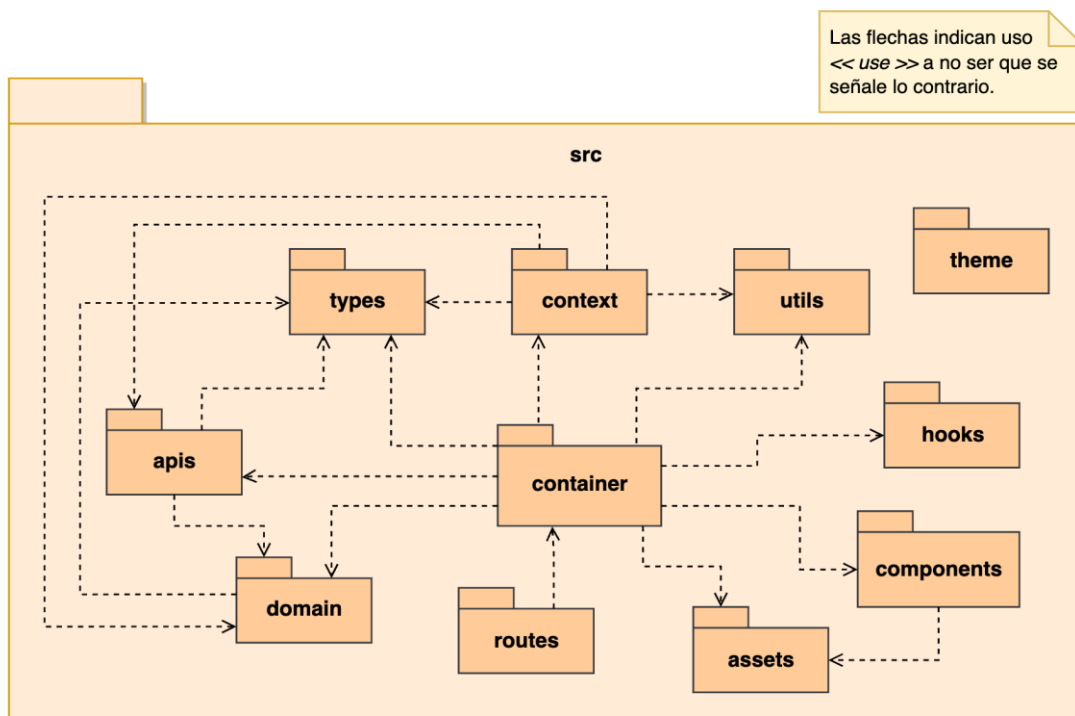


Figura 18 - Descomposición del módulo Webapp

El contexto es clave para el manejo de estados dentro de la aplicación y su relación con los métodos que llaman a los servicios permiten, en la mayoría de los casos, limitar la lógica del componente a activar acciones provistas por el contexto y estar observando los cambios.

Se decidió separar los componentes personalizables como diálogos y botones que pudieran ser customizados mediante propiedades, en la carpeta components, de los

componentes con lógica, que utilizan dichos componentes customizados en la carpeta container. De todas formas, la mayor división se realiza dentro de las carpetas anteriores, con subcarpetas que dividen funcionalidades relacionadas en componentes más pequeños.

A continuación, listaremos los elementos utilizados:

- **assets:** Contiene los iconos utilizados en la aplicación web.
- **apis:** Comprende los métodos asincrónicos que permiten la conexión con la API y AWS Cognito.
- **components:** Incluye los componentes personalizados con estilos y propiedades configurables, no tienen lógica propia.
- **container:** Engloba los componentes que tienen lógica de negocio y utilizan componentes propios, de terceros o embeben otros componentes con lógica.
- **context:** Incluye el contexto de la aplicación utilizando Redux y las acciones que repercuten en el estado del mismo.
- **domain:** Contiene las clases del dominio como modelos de entrada y salida para manejar las respuestas y consultas a la API.
- **hooks:** Contiene los hooks customizados creados para solucionar problemas de forma más eficiente tales como useForm.
- **routes:** Comprende las rutas de la aplicación, pudiendo ser con o sin autorización.
- **theme:** Incluye el tema de la aplicación, con los colores primarios, secundarios, entre otros que utilizan todos los componentes para mantener un estilo uniforme.
- **types:** Contiene distintos tipados utilizados a lo largo de toda la aplicación web.
- **utils:** Engloba los métodos utilizados para realizar cálculos o procedimientos a lo largo de toda la aplicación. A su vez, contiene constantes y todos los

textos mostrados al usuario en idioma español, para poder cambiar de idioma fácilmente utilizando una librería específica.

7.6.2. Flujo de publicación de un nuevo curso

Seleccionamos un diagrama de secuencia para mostrar el flujo de publicación de un nuevo curso con videos, ya que es el proceso más complejo de nuestra plataforma (Figura 19).

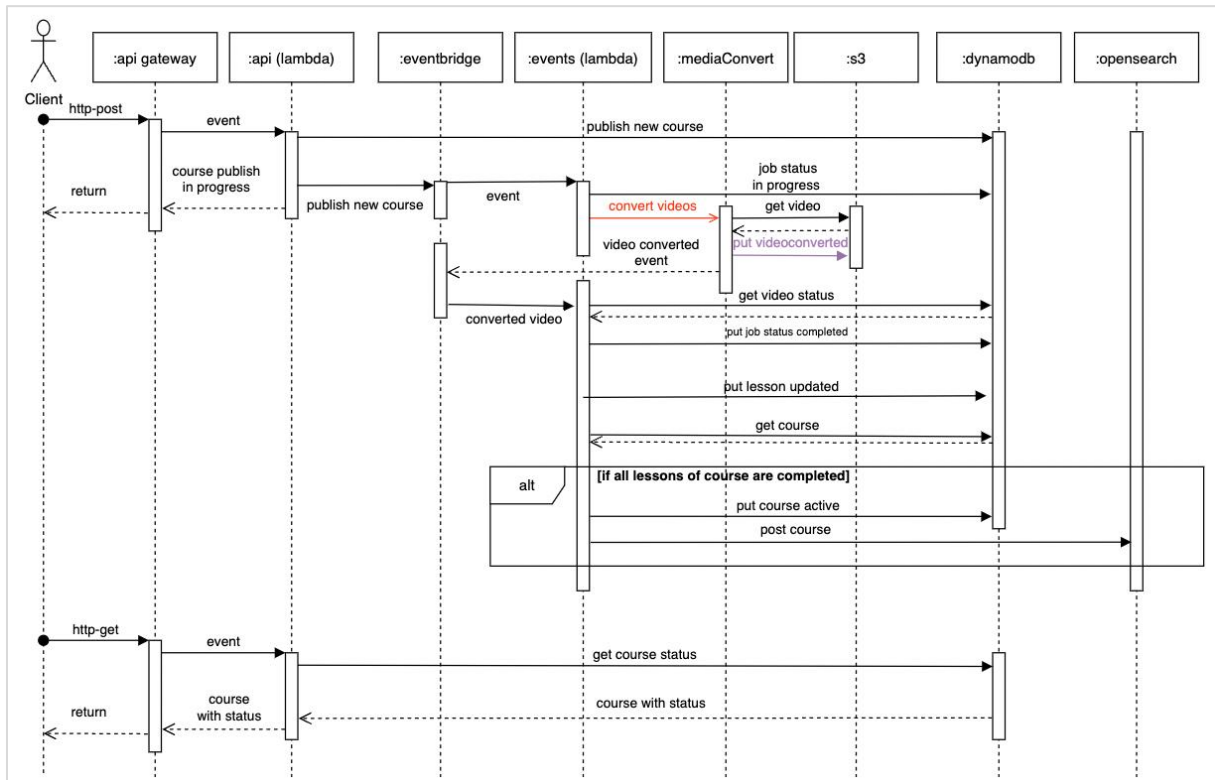


Figura 19 - Diagrama de secuencia para la publicación de un curso

Es importante aclarar que, en este caso, para simplificar el diagrama se optó por evitar los logs generados y guardados en AWS CloudWatch. Asimismo, se tiene en cuenta que el curso no existe, ya que, en caso de existir, se deberían eliminar los videos convertidos anteriormente si fueron cambiados, pero esto no aporta valor y complejiza el entendimiento del flujo en sí. Dicho flujo se rige a su vez por el patrón Async Transaction ya que permite consultar por el estado del curso mientras se está publicando como se puede ver al final del diagrama. [28]

7.7. Conclusiones

Diseñar la arquitectura de un proyecto emprendedor, es un desafío para cualquier equipo de desarrollo. En nuestra experiencia, podemos afirmar que uno de los principales objetivos del diseño de una arquitectura inicial, es establecer las bases para comenzar el desarrollo del proyecto.

Sin embargo, esta tiene que ser capaz de soportar cambios durante la etapa de construcción, de modo que el progreso obtenido hasta el momento pueda ser reutilizado. El uso de herramientas como Serverless Framework y su integración con los servicios de AWS, permitieron obtener una gran versatilidad al ocurrir cambios de esta índole.

Podemos afirmar entonces que la arquitectura diseñada, cumple con los requerimientos, pero principalmente permitirá en un futuro adaptarse a las modificaciones de una manera rápida y eficiente, en caso de ser necesario.

8. Gestión de proyecto

Este capítulo se basa en evidenciar la gestión del proyecto llevada a cabo. Aquí destacaremos evidencia de la planificación y seguimiento de etapas realizadas durante el proceso, como validaciones con interesados, métricas y plan de Releases con sus respectivos Sprints.

También mencionaremos las herramientas utilizadas para la gestión del proyecto, la planificación anual realizada utilizando líneas de tiempo, indicando etapas previas y posteriores al desarrollo. A su vez, mencionaremos la gestión de la comunicación y gestión de riesgos.

8.1. Planificación del proyecto

En esta sección, mencionaremos que planificación de etapas llevamos a cabo para el proyecto, así como la cantidad de tiempo dedicada a cada una de ellas.

Una de las tareas que realizamos en primer lugar, fue la de crear una planificación anual acerca de las etapas en que iba a dividirse nuestro proyecto, tomando como punto de partida la investigación previa que realizamos, y como punto final, la entrega del proyecto.

A continuación, mostramos la planeación inicial y la final, donde los principales cambios radican en los tiempos establecidos para cada etapa, así como la aparición de las etapas de gestión y una segunda etapa de validación con interesados durante el desarrollo (Figura 20 y Figura 21).

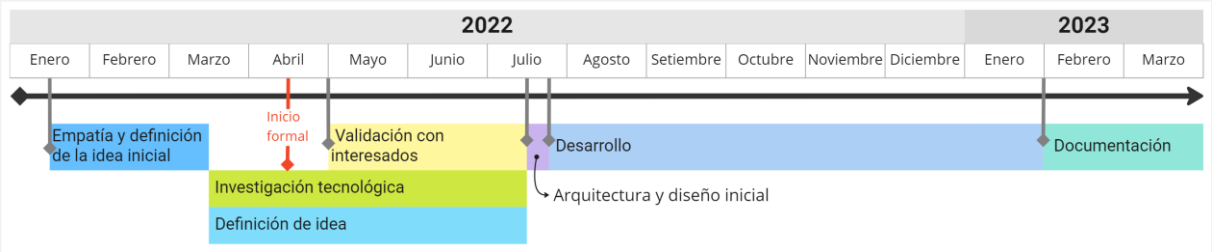


Figura 20 - Planeación anual inicial del proyecto

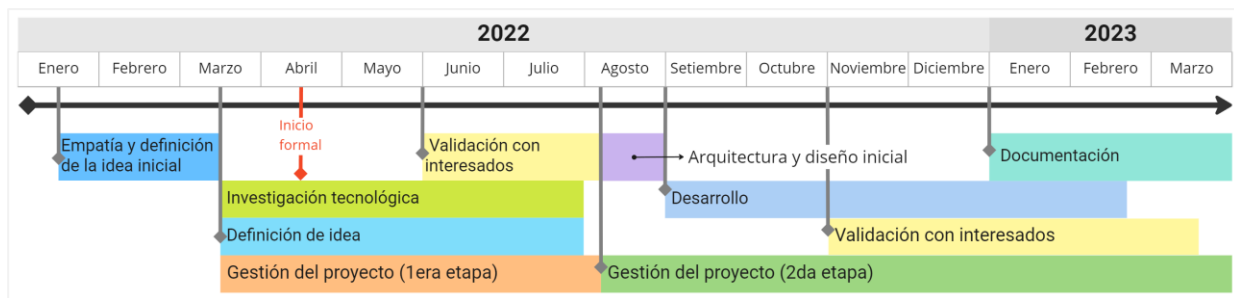


Figura 21 - Planeación anual final del proyecto

Es importante destacar que, durante los primeros meses del proyecto, fueron surgiendo modificaciones a la planeación en cuanto a la duración de etapas y definición de nuevas etapas, las cuales concluyen en el plan final presentado en la Figura 21. Para visualizar las distintas versiones del plan ver [13.14. Anexo 14: Versiones de la planificación anual.](#)

Si bien la fecha de comienzo formal del proyecto fue en abril, tuvimos una etapa inicial en donde, a través de técnicas para empatizar como las entrevistas, relevamos datos provenientes de trabajadores del rubro de la construcción. Aquí también llevamos a cabo un análisis del mercado y de nuestros competidores, así como la creación de un modelo de negocio Canvas, lo cual detallamos a lo largo de la sección [4. Ideación inicial.](#)

Esta etapa fue muy importante para poder tener una primera definición de nuestra idea, la cual posteriormente fuimos puliendo mediante análisis de los resultados obtenidos en las instancias de validación con los interesados.

A mediados de marzo, comenzamos a realizar una investigación enfocada a la parte tecnológica de nuestro producto, en conjunto con la refinación de nuestra idea de modo que estuviera más definida y alineada con lo que queríamos conseguir con el mismo. Esta etapa duró varios meses ya que fue donde definimos nuestros objetivos, metodologías, tecnologías y demás herramientas a utilizar, así como proyecciones de metas o hitos a conseguir a lo largo del proyecto. También pasamos por un largo proceso para definir los requerimientos, realizando múltiples instancias de validación para asegurarnos que fueran los adecuados y los que aportaran mayor valor al producto (como explicamos en la sección [6.1. Proceso de definición de requerimientos](#)).

Todas las etapas anteriores comprendieron lo que denominamos en nuestra planeación, primera etapa de gestión del proyecto, ya que las mismas se enfocaban a la parte de definición, investigación y validación de la idea. Finalizadas dichas etapas, obtuvimos como resultado los RF, RNF, tecnologías a utilizar y una definición de la idea más clara. Esto nos permitió comenzar con lo que llamamos segunda etapa de gestión del proyecto, la cual estaba enfocada a la administración de la construcción, tanto a nivel de software como de documentación a presentar.

Al iniciar esta segunda fase, tuvimos una etapa en donde confeccionamos la arquitectura y diseño, basándonos en lo recolectado y discutido en la investigación realizada en la etapa anterior, también teniendo en cuenta la planificación de hitos que queríamos conseguir y la fecha límite del proyecto. A su vez, se inicializaron y configuraron los ambientes de desarrollo, junto con los servicios de AWS. Asimismo, definimos el plan de Releases que llevaríamos a cabo.

Esta etapa de arquitectura llevó más de lo planeado debido a que surgieron inconvenientes con respecto a las cuentas de AWS y permisos brindados por la Universidad ORT, que no nos permitían acceder a ciertas funcionalidades de administrador. A su vez, no incluían la disposición de determinados servicios, como AWS MediaConvert, clave para la conversión de video. Por esto y siendo que nuestro proyecto es de tipo emprendedor, nos postulamos para obtener créditos de AWS con el fin de poder configurar los servicios con mayor libertad. [29] Este problema y la reconfiguración de ambientes hicieron que esta etapa nos tomara dos semanas más de lo planificado, abarcando un total de tres semanas.

Seguidamente, comenzamos con la etapa de desarrollo (construcción) del producto, la cual se basó fuertemente en el plan de Releases construido en la etapa de arquitectura. Esta etapa de desarrollo fue la que tuvo mayor duración dentro del proyecto, ya que comprendía la realización de todos los RF y RNF definidos. A su vez, efectuamos en paralelo la etapa de validación, donde tuvimos recurrentes instancias con los interesados. Esta etapa duró un poco más que la de construcción debido a que al finalizar el desarrollo, necesitábamos más días para validar resultados

Finalmente, dispusimos de una etapa final dedicada a la documentación, la cual se comenzó a efectuar en paralelo con el desarrollo.

8.2. Herramientas utilizadas

Para poder realizar la gestión del proyecto, las herramientas que usamos fueron Jira, Miro, Google Drive y Microsoft OneDrive.

Jira fue útil para la creación y organización de tareas a cumplir, mediante el uso de un tablero donde plasmamos los requerimientos y los subdividimos en tareas más pequeñas para poder administrarnos de la mejor manera posible (Figura 22).

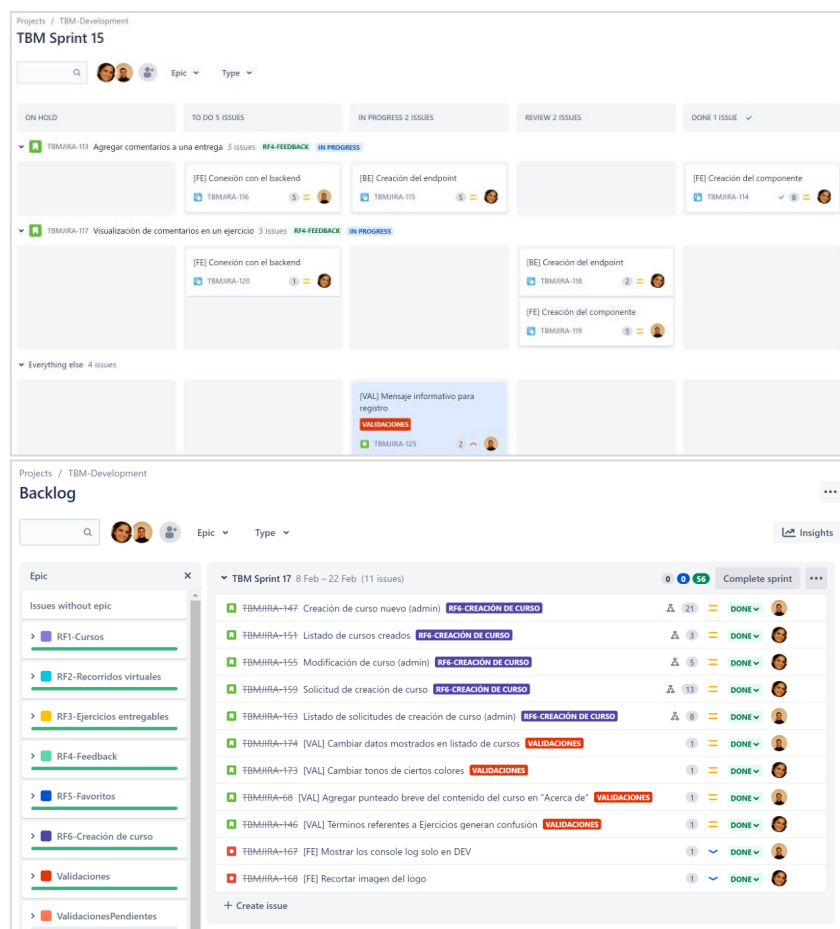


Figura 22 - Uso de Jira

Miro lo utilizamos para la creación de esquemas, planes de Releases, etapas a realizar, tablas con información, notas, entre otros (Figura 23). Google Drive y OneDrive los usamos para almacenar contenido que nos resultara útil, tales como esquemas, resultados de validaciones, registro de horas de trabajo, entre otros (Figura 24).

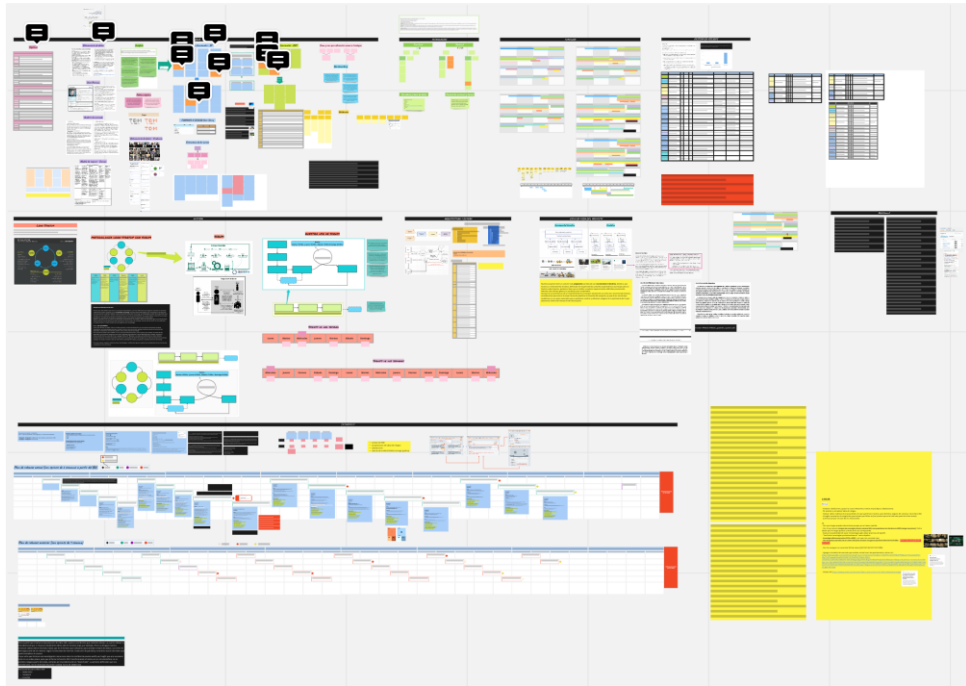


Figura 23 - Uso de Miro

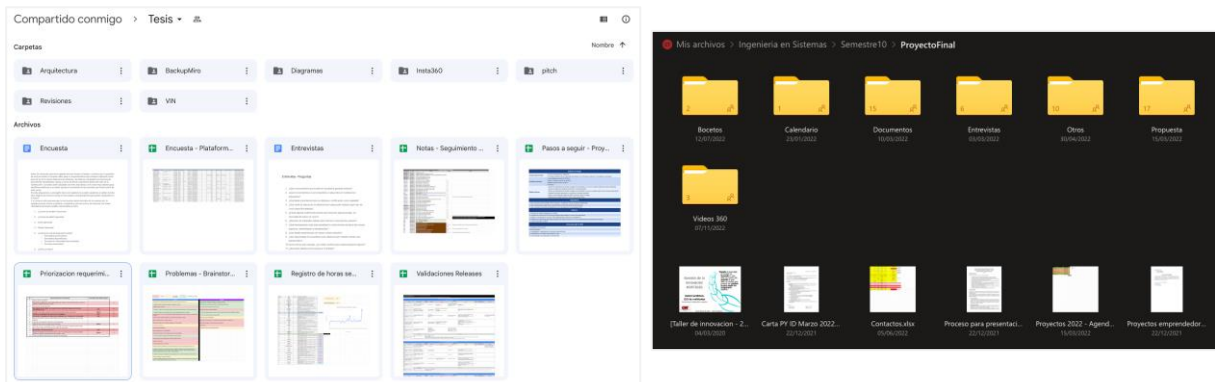


Figura 24 - Uso de Google Drive y OneDrive

8.3. Metodologías de trabajo

8.3.1. Seguimiento del trabajo

Como primer paso, teniendo en consideración todas las etapas que planeamos (para más detalle ver sección [8.1. Planificación del proyecto](#)), creamos un tablero en Jira para definir las tareas a alto nivel que debíamos realizar para cada una de las etapas.

Debido a que esto era para controlar y estar al tanto de las tareas pendientes desde un punto de vista administrativo y de organización interna del equipo, creamos tareas relacionadas a cada etapa. Para poder distinguir a que etapa correspondía cada tarea, utilizamos la funcionalidad de creación de épicas otorgada por Jira para para la

agrupación de tareas en dichas etapas, pudiendo visualizarlas como etiquetas en el tablero. Asimismo, utilizamos distintas columnas para ir colocando las tareas y reflejar su estado, hasta alcanzar finalmente su cumplimiento (Figura 25).

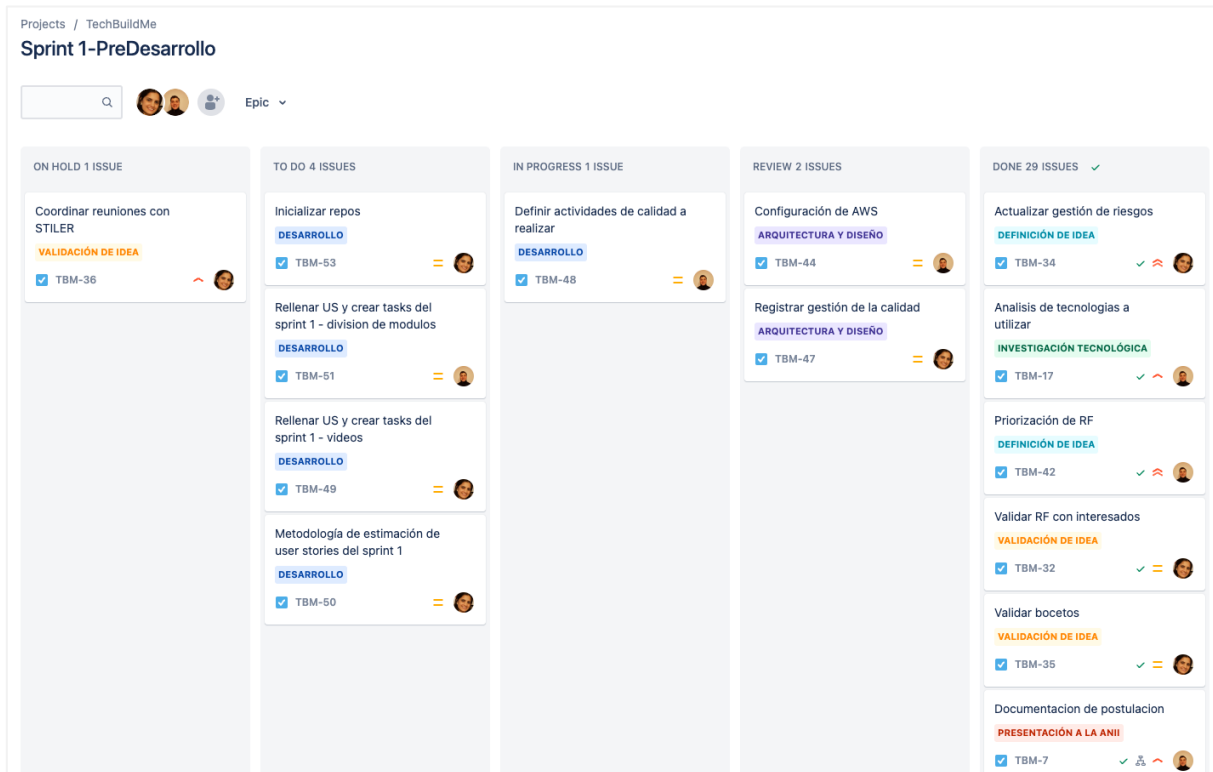


Figura 25 - Tablero en Jira para las distintas etapas

Si bien consideramos todas las etapas planificadas para la utilización del tablero, el mismo fue especialmente útil y relevante durante la primera etapa de gestión del proyecto, siendo este clave para el seguimiento acerca del estado de las actividades, tanto de las finalizadas como las pendientes y en curso.

Queremos destacar que a la par de este proyecto, especialmente durante los primeros meses, estuvimos en contacto con el CIE con el fin de refinar aspectos enfocados al emprendimiento, para poder postularnos a la ANII y así poder obtener apoyo financiero para hacer crecer al mismo. Sin embargo, debido a que nuestro enfoque estaba en el proyecto académico, decidimos dejar estas actividades para una etapa posterior a la entrega de dicho proyecto.

8.3.2. Planificación y seguimiento de la etapa de construcción

Para el caso de la etapa de construcción en particular, como mencionamos anteriormente en la sección [5.3. Metodologías de trabajo](#), hicimos uso de Scrum modificando ciertos aspectos.

En primer lugar, creamos otro tablero en Jira donde agregamos los RF representados como épicas en el mismo (para más detalle ver sección [6.2. Requerimientos funcionales](#)). Luego, elaboramos tareas (User Stories) dentro de cada uno de ellos, desglosándolos en acciones más pequeñas para poder repartirlas entre nosotros y así trabajar en paralelo haciendo un uso más eficiente de nuestro tiempo (Figura 26).



Figura 26 - Épica y su repartición en tareas más pequeñas

Para el caso del feedback obtenido de las validaciones, creamos tareas bajo una épica "Validaciones", de modo que estuvieran todas agrupadas. Las tareas se iban creando en el Backlog y en cada Planning, decidíamos cuales debían realizarse en el siguiente Sprint, en concordancia con el RF que estaba siendo desarrollado.

En adición, para poder distinguir el enfoque de cada User Story y tareas, definimos la siguiente nomenclatura, agregando el texto correspondiente a cada una antes del título:

- [FE] – Frontend.
- [BE] – Backend.

- [CD] – Cloud (AWS/GitHub).
- [VAL] – Proveniente de las validaciones.
- Para el caso de los bugs que se iban encontrando, se cambiaba el tipo de tarea en el Jira directamente.

En la siguiente imagen (Figura 27) puede verse el tablero de Jira, apreciándose algunos de los puntos mencionados anteriormente.

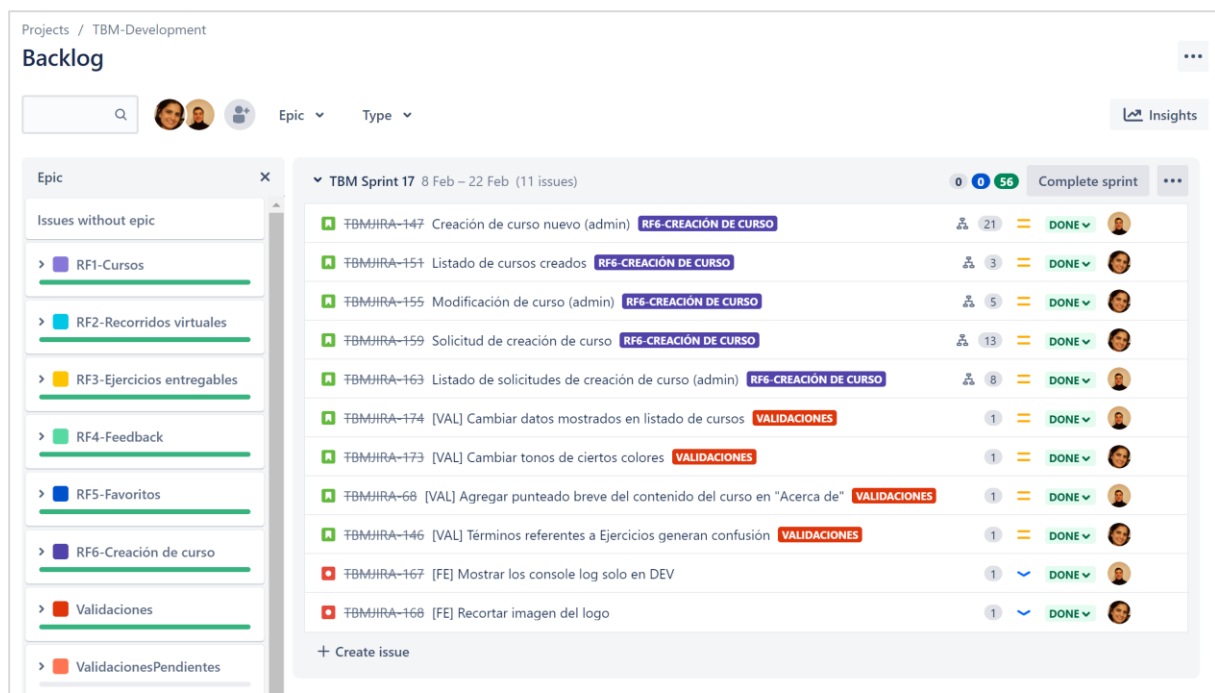


Figura 27 - Épicas y algunas de las User Stories y tareas de la etapa de construcción

En cuanto a la duración de los Sprints, inicialmente fueron de una semana. Sin embargo, luego de transcurrir los primeros, notamos que el tiempo era muy corto considerando las tareas que teníamos asignadas y las ceremonias que realizábamos, las cuales nos consumían tiempo de desarrollo. Por estos motivos, a partir del octavo Sprint decidimos cambiar la duración a dos semanas, pudiendo así tener mejores resultados y métricas más útiles para medir nuestro avance.

Una vez comenzada un Sprint, movíamos las tareas que habíamos seleccionado previamente en la Planning del Backlog al Sprint actual. Para saber el estado de cada una de ellas, utilizamos las mismas columnas que mencionamos en la parte anterior (Figura 28).

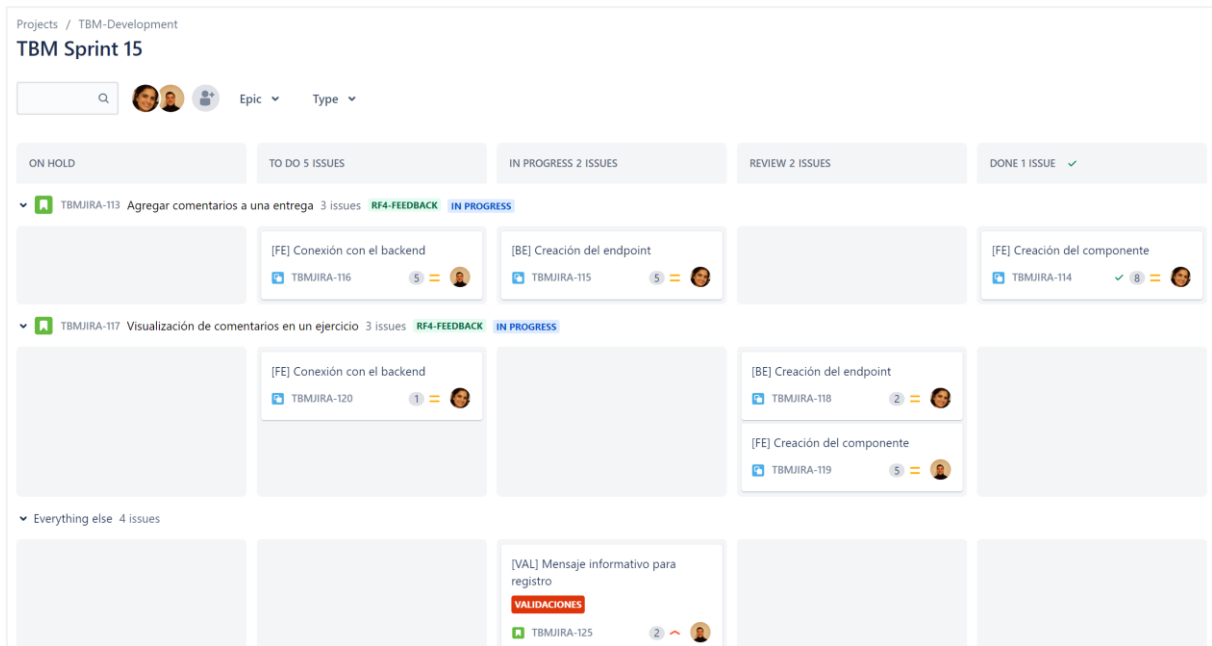


Figura 28 - Tablero de Jira para la etapa de desarrollo

8.3.3. Estimación de las tareas

Respecto a la estimación, decidimos utilizar la escala de Fibonacci para asignar los Story Points a las tareas. Para ello, tuvimos en cuenta la duración de los Sprints, las horas diarias dedicadas por integrante, la división de tareas y el uso de Fibonacci como estándar en la industria, definiéndola como se muestra en la Tabla 7.

Story Points	Equivalente en días
1	Menor a 1 día
2	1 día
3	1 a 2 días
5	2 a 3 días
8	3 a 5 días
13	5 a 7 días
21	7 a 10 días
34	10 a 14 días

Tabla 7 - Escala de Story Points (Fibonacci)

8.4. Plan de Releases

En primer lugar, definimos las condiciones que debía cumplir un Release para considerarse como tal, las cuales eran:

- la finalización completa de un RF, es decir, la terminación de todas las tareas que formaban un RF.
- código testeado y funcional de manera correcta en la rama principal (master).
- las validaciones con interesados no estaban incluidas, sino que estas se realizaban en el Sprint siguiente, en paralelo con el desarrollo del mismo. Esto se debía a que los interesados contaban con horarios acotados y coordinar reuniones con ellos no era inmediato, por lo que consideramos que esperar a tener todas las validaciones iba a retrasarnos, corriendo el riesgo de no llegar a tiempo con el alcance planificado.

En base a lo anterior y para poder llevar a cabo la etapa de construcción del proyecto lo más eficientemente posible, creamos un plan de Releases. Para ello, en base a cada Release definido y que representaba, considerábamos cuantos Sprints necesitábamos para construirlo (Figura 29).

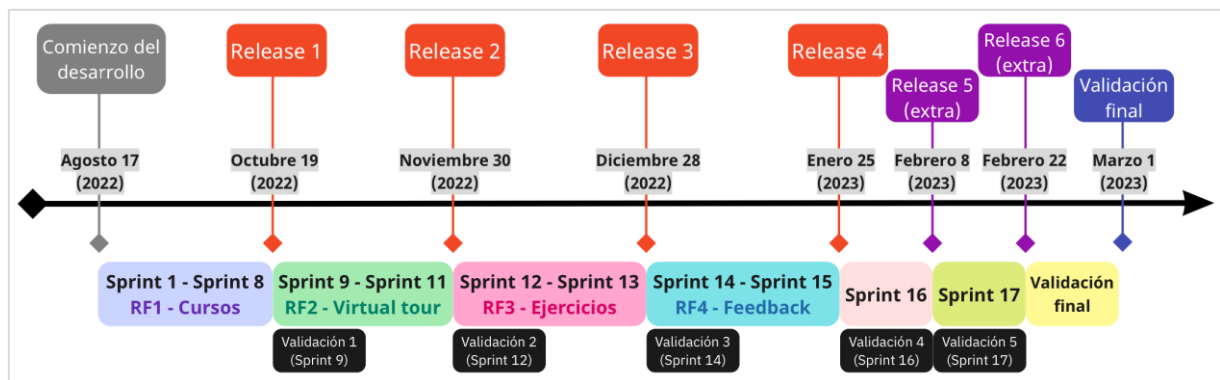


Figura 29 - Plan de Releases

Como mencionamos anteriormente, cada Release implicaba la finalización de un RF, y por lo tanto, de todas las tareas contenidas en él (para ver el listado de RF, ver sección [6.2. Requerimientos funcionales](#)).

Como se puede observar en la imagen anterior, el primer Release correspondía al RF1, relacionado a la visualización de cursos, el cual ocupó los primeros ocho Sprints.

Es importante destacar que el RF1 estaba planificado para finalizarse el día 5 de octubre de 2022, pero como mencionamos en la sección [8.1. Planificación del proyecto](#), las dos semanas extra que se utilizaron en la etapa de arquitectura y diseño correspondieron al Sprint 1 y 2. Por este motivo, más los inconvenientes mencionados en la sección [8.3.2. Planificación y seguimiento de la etapa de construcción](#) acerca de la duración de los Sprints, el Release 1 fue obtenido el día 19 de octubre de 2022, desplazando dos semanas los posteriores Releases.

Como nota importante a tener en cuenta, la transición de la duración de los Sprints de una a dos semanas no afectó la duración de ningún Release. Esto debido a que, sin contar los Sprints estimados para el Release 1, el resto de ellos contaba con una cantidad par de Sprints estimados, haciendo sencillo este cambio de duración.

El segundo Release refería al RF2 de recorridos virtuales y así en adelante con el resto. Por último, contábamos con dos Releases adicionales, en los cuales implementamos dos RF más, considerados no prioritarios pero que complementaban a las funcionalidades ya implementadas.

Para cerrar la etapa de construcción, definimos una última instancia de validación con los interesados, de modo de obtener sus comentarios acerca de nuestro producto en su forma más completa. Las validaciones se pueden ver más detalladas en la sección [10.3.3. Releases](#).

A continuación, mostramos una tabla resumiendo la fecha de inicio y fin de cada RF y cuantos Sprints llevó (Tabla 8). Vale aclarar que para el RF1 se contaron como si los Sprint hubieran sido de dos semanas para simplificar los datos. Esto sin contar el Sprint 1 y 2 por lo mencionado anteriormente.

Release	RF	Inicio	Fin	Sprints realizados
1	RF1 – Cursos	31/08/22	19/10/22	3 y ½
2	RF2 – Recorridos virtuales	20/10/22	30/11/22	3
3	RF3 – Ejercicios entregables	01/12/22	28/12/22	2
4	RF4 – Feedback	29/12/22	25/01/23	2
5	RF5 – Favoritos	26/01/23	08/02/23	1
6	RF6 – Creación de curso	09/02/23	22/02/23	1

Tabla 8 - Tabla simplificada de Releases

Para ver en detalle el contenido específico de cada Sprint, ir a [13.15. Anexo 15: Detalle de Sprints](#).

8.5. Principales hitos del proyecto

Como hitos a destacar durante el transcurso del proyecto tenemos los siguientes:

- **Entrega de anteproyecto**

Esta fue la primera entrega formal de documentos conteniendo información detallada acerca de nuestro proyecto, incluyendo datos relevantes, descripción, motivación, así como el proceso de relevamiento de datos, análisis de mercado y competidores y demás. Luego de realizar esta entrega tuvimos que esperar por la aprobación de nuestra idea, por lo cual representa un momento importante en nuestro proyecto y que dio comienzo al mismo una vez recibida la respuesta positiva.

- **Primera reunión con la tutora**

Esta instancia fue de gran importancia para nosotros, ya que conocimos a quien iba a guiarnos y apoyarnos a lo largo del proyecto desde el punto de vista académico, buscando que lo lleváramos adelante lo mejor posible y brindándonos sus consejos basados en su experiencia. Además, aquí realizamos el primer intercambio de opiniones acerca de cómo íbamos a organizarnos y en qué aspectos debíamos enfocarnos más.

- **Entrega de contrato de cesión de derechos**

Este documento era relevante ya que definía las condiciones y derechos del proyecto una vez finalizado el mismo. Además, teníamos un plazo definido para poder entregarlo revisado y firmado, por lo que nos tomamos el tiempo para analizarlo y asegurarnos de estar de acuerdo con lo declarado en el contrato.

- **Validación de prototipos**

Esta instancia fue de suma importancia, ya que los prototipos representaban una primera versión de nuestra aplicación, y el obtener las opiniones de los interesados era necesaria para asegurarnos que nos estábamos enfocando en lo correcto y que aportaba valor al sector.

- **Definición de requerimientos**

Todo el proceso de obtención de requerimientos fue vital para tener definidas las funcionalidades que íbamos a implementar y que esfuerzo requerirían. Asimismo, nos ayudó a tener un orden y una visión más clara en cuanto a proyecciones sobre la organización de tareas a lo largo de la etapa de construcción.

- **Releases**

Cada Release supuso un hito porque representaban la finalización de un RF en su totalidad, significando que esa funcionalidad ya estaba en condiciones para poder ser validada con los interesados.

- **Validación de cada Release**

En estas instancias fue donde pudimos obtener muchos comentarios útiles y posibles mejoras de parte de los interesados. Considerando que eran parte de nuestro público objetivo, resultaba muy importante llevar a cabo estas reuniones una vez obtenido un Release. Esto porque nos permitían validar lo que estábamos haciendo con personas con experiencia en el rubro y poder verificar que nuestra aplicación aportara valor.

Asimismo, la instancia de validación del Release 4 en particular fue más significativa ya que representó la validación del MVP definido.

- **Validación final**

Similar a lo mencionado en el punto anterior, esta última instancia de validación representó uno de los mayores hitos de nuestro proyecto. Aquí obtuvimos feedback de los interesados acerca de una versión mucho más completa que las anteriores, permitiéndoles apreciar la aplicación con todas las funcionalidades planeadas implementadas.

- **Entrega final**

El mayor hito de nuestro proyecto ya que representa la finalización de una gran etapa académica para nosotros. Para esta fecha debíamos entregar la documentación completa, asegurándonos previamente de haber plasmado correctamente todo el proceso transcurrido durante el proyecto.

8.6. Métricas

A continuación, presentamos las distintas métricas obtenidas a lo largo del proyecto.

8.6.1. Gestión

- **Distribución de horas de trabajo**

Desde los inicios del proyecto utilizamos una planilla para el registro de horas diarias, especificando en detalle las actividades realizadas por semana y las horas dedicadas por integrante (para más detalle ver [13.16. Anexo 16: Registro detallado de horas de trabajo](#)). A su vez, fuimos detallando en la etapa de construcción, las horas dedicadas por Sprint.

La siguiente tabla (Figura 30) refleja la dedicación de horas, agrupada según la información que teníamos registrada, incluyendo fecha de inicio y fin, total de horas dedicadas y su correspondiente porcentaje.

Descripción	Inicio	Fin	Horas	Dedicación
Proyecto	17-04-22	30-03-23	1626	100.0%
Definición de la idea, investigación tecnológica y validación	17-04-22	10-08-22	488	30.0%
Arquitectura y diseño (inicial)	10-08-22	01-09-22	88	5.4%
Desarrollo	01-09-22	22-02-23	640	39.4%
Reuniones/Ceremonias y QA (durante el desarrollo)	10-08-22	22-02-23	178	10.9%
Validaciones releases	20-10-22	23-03-23	30	1.8%
Documentación	01-01-23	30-03-23	202	12.4%

Figura 30 - Dedicación de horas durante el proyecto

Asimismo, a modo de simplificar la visualización de la distribución de horas dedicadas a las etapas, creamos un gráfico de torta para ilustrar los porcentajes antes mencionados, el cual puede apreciarse en la Figura 31.

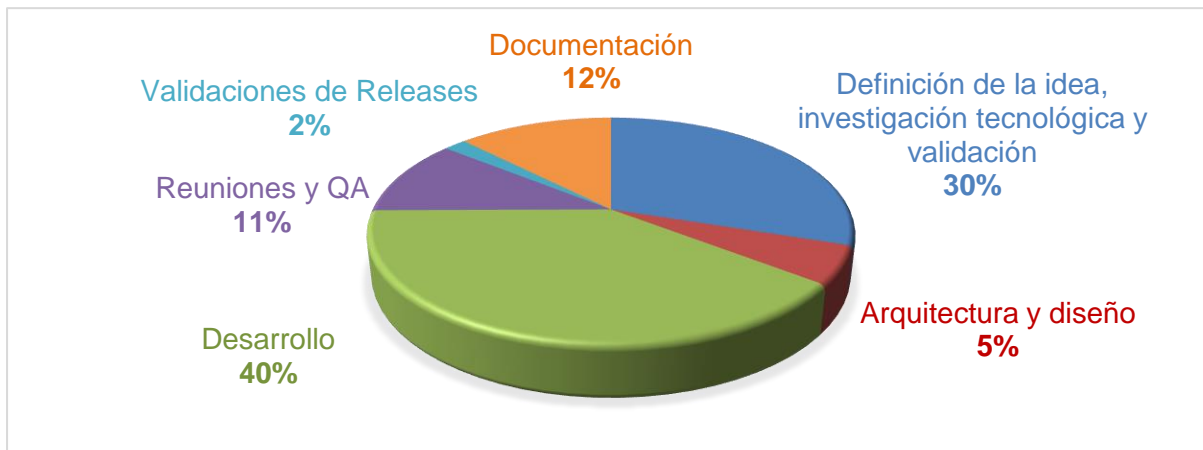


Figura 31 - Gráfico simplificado de dedicación de horas del proyecto

Para la segunda etapa de gestión del proyecto, utilizando la planilla de registro de horas diarias mencionada antes, la dedicación por Sprint y las validaciones realizadas, pudimos desglosar de mejor manera la dedicación concerniente a esta segunda etapa.

En este caso separamos la arquitectura y diseño inicial, el desarrollo, las reuniones o ceremonias en conjunto con las actividades de QA realizadas por el equipo durante el desarrollo, las validaciones con los interesados y el tiempo dedicado a la documentación.

Sin embargo, para la primera etapa de gestión del proyecto, debido a que el equipo no tenía práctica en cuanto al registro de horas y las actividades realizadas se entrelazaban mucho, el registrar de forma separada las etapas correspondientes a esta primera parte de gestión resultaba confuso y poco exacto, por lo que optamos por agrupar esas etapas.

En primer lugar, lo que más tiempo ocupó fue la etapa de desarrollo. A su vez, si bien señalamos a la arquitectura y diseño como una etapa previa y separada a la de desarrollo, luego continuó dentro de esta última, ya que la arquitectura inicial fue siendo modificada para adaptarse al feedback que íbamos recibiendo por parte de los interesados en la etapa de validación.

Destacamos también, junto con las etapas mencionadas anteriormente, las reuniones/ceremonias y actividades de QA, ya que nos permitieron gestionar eficientemente el desarrollo.

En segundo lugar, tenemos el conjunto de las etapas de definición de la idea, investigación tecnológica relacionada e instancias de validación, como las de prototipos, RF y RNF. El hecho de que nuestro proyecto sea un emprendimiento fue el principal motivo por el cual dedicamos esa cantidad de tiempo en estas etapas. Debíamos asegurarnos de relevar los datos y hacer los análisis correspondientes, para luego crear prototipos que validaríamos para poder definir nuestros requerimientos de la mejor manera posible, dependiendo considerablemente de nuestros interesados para obtener feedback de estas validaciones. Asimismo, tuvimos que definir nuestros objetivos, que metodologías eran las más adecuadas, planificar y proyectar hitos y seleccionar las tecnologías a utilizar, entre otras actividades.

Por último, la etapa relacionada a la documentación tuvo inicio a principios de 2023, de modo que pudiéramos planificar su estructura con tiempo para que la información fuera plasmada de forma correcta, a la vez que dábamos un cierre a la construcción del producto.

Es importante destacar que la validación fue una de las etapas que estuvo presente en la mayor parte de nuestro proyecto, siendo una de las fundamentales, ya que es donde obteníamos el feedback de los interesados acerca de nuestra aplicación. Debido a que no teníamos un cliente establecido, la continuidad de estas instancias eran clave para guiarnos en el desarrollo de la misma.

8.6.2. Construcción

- **Estimado vs. Real de Story Points**

A continuación, mostramos una gráfica representando los Story Points estimados en comparación frente a los realizados para cada Sprint (Figura 32).

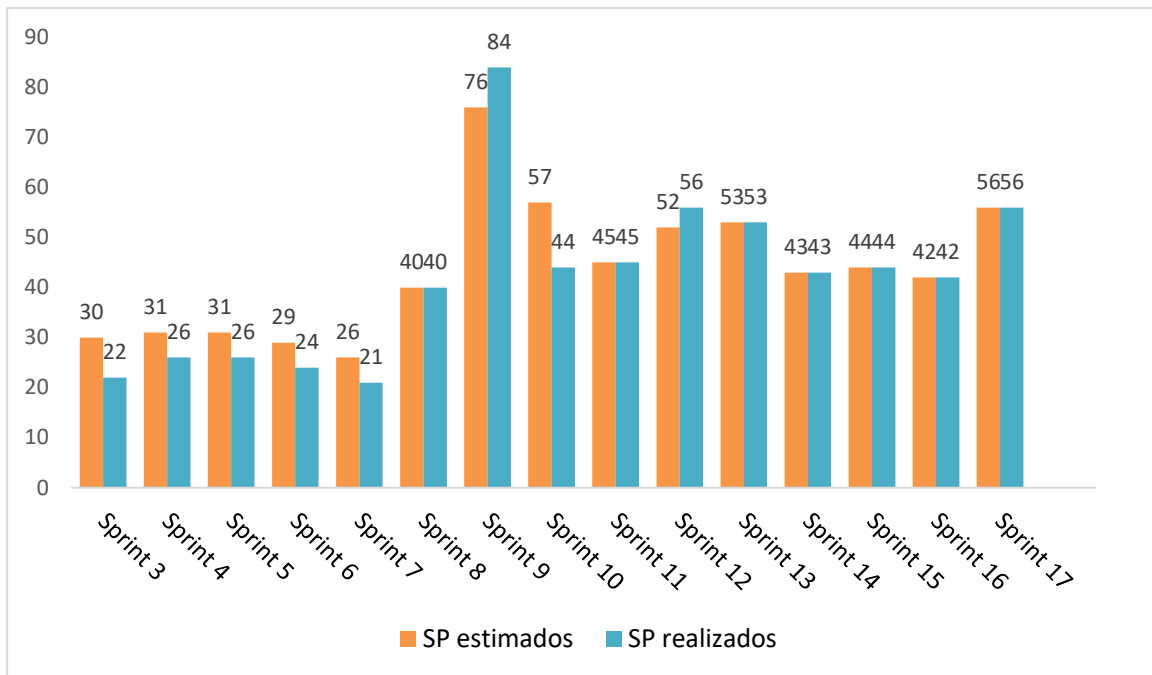


Figura 32 - Estimado vs. real de SP por Sprint

Vale destacar que el no contemplar los Sprint 1 y 2 y el no cumplir con todos los SP planificados desde el Sprint 3 al 7, se relaciona con los problemas de configuración de AWS y cambios de duración mencionados en la sección [8.4. Plan de Releases](#). Asimismo, las tareas no realizadas del sprint 10 corresponden a bugs encontrados y planificados que se atacaron en el Sprint siguiente.

En el caso de Sprint 9 y 12, hicimos más SP de los planeados debido a que contábamos con más tiempo fuera del compromiso establecido para realizar bugs que habíamos detectado.

Estos datos nos permitieron planificar con mayor exactitud las tareas a realizar en cada sprint. Algo importante a resaltar es que, durante el desarrollo del mismo, podíamos agregarle tareas y/o bugs, ya que la metodología Scrum lo permite.

Para ver más en detalle la composición de cada Sprint, ir a [13.15. Anexo 15: Detalle de Sprints](#).

- **Velocidad del equipo**

A continuación, mostramos la gráfica correspondiente al cálculo de la velocidad por Sprint del equipo (Figura 33).

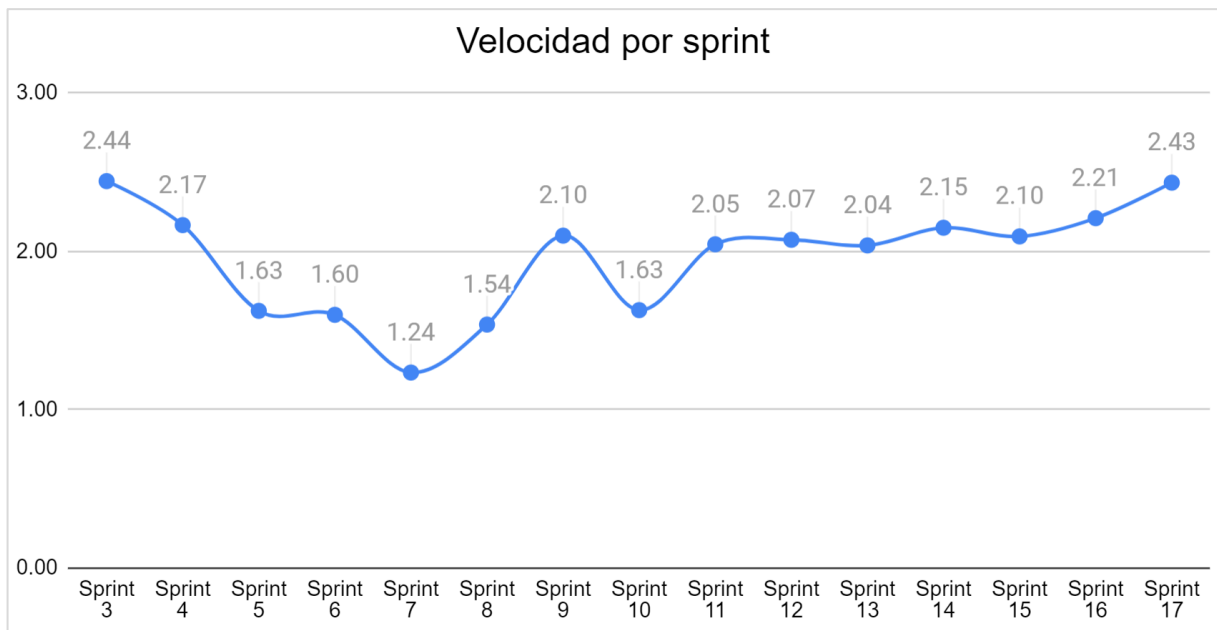


Figura 33 - Velocidad por Sprint

Como se puede observar en la gráfica, en los primeros Sprints la velocidad del equipo se vio disminuida cada vez más. Esto se debió a la realización de ceremonias con Sprint de duración de una semana, como explicamos en la sección [8.3.2. Planificación y seguimiento de la etapa de construcción](#). Una vez alterada dicha duración a dos semanas, la velocidad fue aumentando para luego hacerse más constante.

En el Sprint 8 fuimos adaptándonos al cambio, modificando la forma en que estimábamos las tareas, por lo cual se empezó a notar un crecimiento. Es importante destacar que, a pesar de contar con dos semanas, el equipo se estaba ajustando a los nuevos tiempos, por ello la diferencia con las velocidades posteriores.

Por el contrario, en el Sprint 9 tuvimos un salto en la velocidad, también influyendo en esto el hecho de que ambos integrantes pedimos días en nuestros respectivos trabajos, pudiendo dedicar más tiempo al proyecto.

En el Sprint 10, nuevamente tuvimos una disminución en la velocidad debido a que uno de los integrantes se enfermó, lo cual dificultó trabajar completamente en las tareas que tenía asignadas, no pudiendo completarlas en su totalidad, afectando negativamente a la velocidad del equipo. Cabe aclarar que las tareas no realizadas correspondían a bugs y no a implementación de funcionalidades ni tareas surgidas de las validaciones.

8.7. Gestión de la comunicación

A continuación, detallaremos como gestionamos los distintos tipos de comunicación que tuvimos a lo largo del proyecto.

○ Equipo

Para la comunicación entre nosotros, utilizamos canales como WhatsApp, Google Meet, Zoom y Teams. A su vez mantuvimos Dailies y Plannings para conocer el estado actual de cada uno y organizar las tareas para los Sprints respectivamente. A modo de evaluar el funcionamiento de la metodología, realizamos Sprints Review al finalizar cada Sprint y Sprint Retrospectives al completar todos los Sprints que comprenden un Release.

○ Tutora

Como medios para comunicarnos con la tutora, utilizamos principalmente WhatsApp y Gmail para la comunicación diaria, mientras que para las reuniones, en un principio utilizamos Zoom y en los últimos meses migramos a Teams debido a que no tenía límite de tiempo en las videollamadas.

○ Interesados

En el caso de los interesados, los canales utilizados fueron WhatsApp y Gmail para la coordinación de reuniones, siendo las mismas llevadas a cabo a través de Zoom o Google Meet, dependiendo las preferencias del interesado en cuestión.

8.8. Gestión de riesgos

8.8.1. Detección

Para poder detectar los riesgos, tuvimos varias instancias de análisis durante las etapas iniciales del proyecto (definición de la idea e investigación). Allí dejamos plasmados los riesgos que podían llegar a manifestarse en dichas etapas y en las próximas. Luego, en el caso de las etapas de arquitectura y diseño y la de construcción, reiteramos las instancias de análisis y agregamos más riesgos que identificamos. Asimismo, separamos los riesgos por etapas, pudiendo aparecer en

más de una, como el caso de los relacionados a las validaciones, las cuales son llevadas a cabo en reiteradas ocasiones durante el proyecto.

8.8.2. Estrategia

Con el fin de tener una visión y un control sobre los riesgos del proyecto, ideamos un plan, listando en una tabla cada uno de ellos, junto con su descripción, su magnitud (calculada a partir del impacto y probabilidad de ocurrencia) y plan de respuesta.

Para calcular la magnitud, utilizamos los siguientes criterios:

- **Escala de Impacto**

0 – Ninguno, 1 – Marginal, 2 – Poco importante, 3 – Importante (puede retrasar el proyecto), 4 – Crítica (puede detener el proyecto), 5 – Catastrófica (fracaso del proyecto).

- **Escala de Probabilidad de Ocurrencia**

0.0 – no probable, 0.2 – poco probable, 0.4 – probable, 0.6 – muy probable, 0.8 – altamente probable, 1.0 – se convierte en problema.

- **Estrategias para planes de respuesta**

Las estrategias aplicadas como plan de respuesta se dividen en las siguientes categorías:

Evitar: el equipo del proyecto actúa para eliminar la amenaza o para proteger al proyecto de su impacto, por lo general implica cambiar el plan del proyecto con el fin de eliminar por completo la amenaza.

Transferir: el equipo traslada el impacto de una amenaza a un tercero, junto con la responsabilidad de la respuesta. Esto no lo elimina solo se lo confiere a un tercero.

Mitigar: el equipo de proyecto actúa para reducir la probabilidad de ocurrencia o impacto de un riesgo. Implica reducir a un umbral aceptable la probabilidad y/o el impacto de un riesgo adverso.

Aceptar: el equipo decide reconocer el riesgo y no tomar ninguna medida a menos que se materialice, debido a que no es posible abordarlo de otra manera. La aceptación pasiva no requiere ninguna acción, salvo documentar la estrategia dejando que el equipo del proyecto aborde los riesgos conforme se presentan. La aceptación activa consiste en establecer reservas para manejar los riesgos.

A continuación, mostramos una tabla mencionando los riesgos y sus características, indicando su impacto (I), probabilidad de ocurrencia (PO) y la mayor magnitud (M) que podrían tomar los mismos. A su vez, a lo largo del proyecto los planes de respuesta para cada uno de ellos no variaron.

ID	Riesgo	I	PO	M	Etapas de detección
1	Alcance superado	3	0,4	1,2	Definición de la idea
Especificación					
Los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.					
Plan de respuesta					
Mitigar: como el tiempo es una constante que no podemos modificar, se documentarán las razones por las cuales no se pudo cumplir con todo lo planeado, y se establecerá un "plan B" para afrontar la situación, como priorizar la finalización de los RF y RNF que más valor aporten.					

Tabla 9 - Riesgo detectado 1

ID	Riesgo	I	PO	M	Etapas de detección
2	Falta de integrantes	4	0,6	2,4	Definición de la idea
Especificación					
Uno de los integrantes se enferma o le surgen imprevistos, por lo que no puede aportar al proyecto / no puede cumplir con sus tareas asignadas en tiempo y forma.					
Plan de respuesta					
Mitigar: se revisan las prioridades y se hace enfoque en las tareas más prioritarias, de modo que el integrante restante pueda avanzar en ellas, aunque fueran tareas que no le estaban asignadas.					

Tabla 10 - Riesgo detectado 2

ID	Riesgo	I	PO	M	Etapa de detección
3	Disputas entre integrantes	3	0,4	1,2	Definición de la idea
Especificación					
Se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.					
Plan de respuesta					
Evitar: se estimula la conversación y el buen trato entre los integrantes del equipo. Además, se utilizan mecanismos para poder tener una buena comunicación a distancia, como WhatsApp, Zoom, Google Meet.					

Tabla 11 - Riesgo detectado 3

ID	Riesgo	I	PO	M	Etapa de detección
4	Disputas entre integrantes	4	0,4	1,6	Definición de la idea
Especificación					
El poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.					
Plan de respuesta					
Mitigar: se realizan investigaciones del sector y validaciones recurrentes con interesados que participan activamente en el mismo.					

Tabla 12 - Riesgo detectado 4

ID	Riesgo	I	PO	M	Etapa de detección
5	Tecnologías incorrectas	3	0,2	0,6	Investigación tecnológica
Especificación					
Las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.					
Plan de respuesta					
Evitar: se realiza una investigación previa a la etapa de construcción, lo más amplia posible para hallar las tecnologías que mejor se adecuen al proyecto y descartar las que no sean útiles.					

Tabla 13 - Riesgo detectado 5

ID	Riesgo	I	PO	M	Etapas de detección
6	Interesados en desacuerdo	4	0,2	0,8	Validación con interesados
Especificación					
Los interesados no están de acuerdo con la idea propuesta, por lo que dejan de brindar apoyo.					
Plan de respuesta					
Evitar: se intenta negociar con el interesado para convencerlo de seguir apoyando el proyecto, ofreciendo cambiar algunas cosas que no sean de su agrado y haciendo que se sienta lo más cómodo posible en las interacciones que tenga con el equipo.					

Tabla 14 - Riesgo detectado 6

ID	Riesgo	I	PO	M	Etapas de detección
7	Interesados no disponibles	3	0,8	2,4	Validación con interesados
Especificación					
Los interesados son difíciles de contactar y por lo tanto, coordinar reuniones de validación con ellos.					
Plan de respuesta					
Mitigar: se intenta contactar con los interesados con antelación para cubrir el caso en que cancelen reuniones, para así poder reagendarlas con tiempo. En caso de no poder mitigar y no contar con todas las validaciones, en cuanto a la usabilidad, se podrían tomar algunas de las heurísticas de Nielsen como guía para hacer una evaluación. A su vez, el usar metodologías ágiles como SCRUM, nos permite seguir adelante aún sin tener revisiones, debido al flujo que se maneja.					

Tabla 15 - Riesgo detectado 7

ID	Riesgo	I	PO	M	Etapas de detección
8	Arquitectura cambiante	3	0,4	1,2	Arquitectura y diseño
Especificación					
Debido a la naturaleza de nuestro proyecto, los cambios son inevitables, por lo que la arquitectura puede verse alterada.					
Plan de respuesta					
Mitigar: se toman en cuenta buenas prácticas, patrones y principios de diseño para que la arquitectura quede definida lo mejor posible, de modo que si se deben realizar cambios, tengan un impacto menor.					

Tabla 16 - Riesgo detectado 8

ID	Riesgo	I	PO	M	Etapas de detección
9	Objetivos no alcanzados en sprint	3	0,4	1,2	Desarrollo
Especificación					
Los objetivos del sprint no son alcanzados debido a imprevistos que impidan o retrasen lo planeado, por lo que las tareas estimadas se atrasan.					
Plan de respuesta					
Evitar: si surge algún imprevisto, se agrega para el próximo sprint. A excepción de que sea de suma importancia y su realización impida el avance del proyecto.					

Tabla 17 - Riesgo detectado 9

ID	Riesgo	I	PO	M	Etapas de detección
10	Problemas con servicios Cloud	4	0,6	2,4	Desarrollo
Especificación					
Alguno de los servicios que utilizamos remotamente falla, prohibiéndonos utilizarlo o configurarlo remotamente.					
Plan de respuesta					
Evitar: en caso de suceder problemas con esos servicios, utilizaremos otro servidor personal que nos permita tener más acceso y configuraciones.					

Tabla 18 - Riesgo detectado 10

ID	Riesgo	I	PO	M	Etapas de detección
11	Mala estimación de la duración de tareas	3	0,4	1,2	Desarrollo
Especificación					
Mala estimación de la duración a la hora de realizar los cambios, haciendo que demoren más de lo previsto.					
Plan de respuesta					
Mitigar: se intenta realizar las tareas dentro del tiempo establecido, respetando el cronograma. Si es posible, se aumentan las horas de trabajo para dedicarles más tiempo.					

Tabla 19 - Riesgo detectado 11

ID	Riesgo	I	PO	M	Etapa de detección
12	No conseguir obras para grabar los recorridos	3	0,2	0,6	Desarrollo
Especificación					
No se consigue acceso a obras de construcción, de modo que no podemos obtener grabaciones realistas.					
Plan de respuesta					
Mitigar: se comunica con tiempo y con varias personas que tengan contacto con el sector, teniendo la mayor cantidad de opciones posible.					

Tabla 20 - Riesgo detectado 12

ID	Riesgo	I	PO	M	Etapa de detección
13	No conseguir cámara 360	3	0,2	0,6	Desarrollo
Especificación					
No se consigue una cámara capaz de grabar videos 360, de modo que no podemos conseguir tener recorridos virtuales en obras realistas.					
Plan de respuesta					
Aceptar (activamente): se documenta el motivo por el cual no se consiguió la cámara 360 y se utilizan otras alternativas u otro tipo de videos para los recorridos virtuales.					

Tabla 21 - Riesgo detectado 13

ID	Riesgo	I	PO	M	Etapa de detección
14	Falta de fondos en AWS	4	0,2	0,8	Desarrollo
Especificación					
Debido a un alto uso de los servicios de AWS, se consume todo el dinero de los fondos académicos que nos otorgaron.					
Plan de respuesta					
Mitigar: se lleva un seguimiento de los gastos realizados por el uso de servicios, de modo que podamos conocer el saldo restante y organizarnos en base a ello.					

Tabla 22 - Riesgo detectado 14

8.8.3. Seguimiento

Con el fin de saber si algún riesgo de la tabla se había manifestado, o si había aparecido alguno nuevo que no había sido detectado aún, cada un mes aproximadamente, nos tomamos un tiempo para revisarlos. En caso de ser necesario, actualizábamos el riesgo existente o creábamos uno nuevo.

Además, creamos una gráfica donde representamos la evolución de los riesgos más relevantes, es decir, aquellos cuya magnitud tomó valores más elevados a lo largo del proyecto (Figura 34). También realizamos la evolución del resto de los riesgos identificados, la cual puede verse en [13.17. Anexo 17: Evolución de los riesgos.](#)

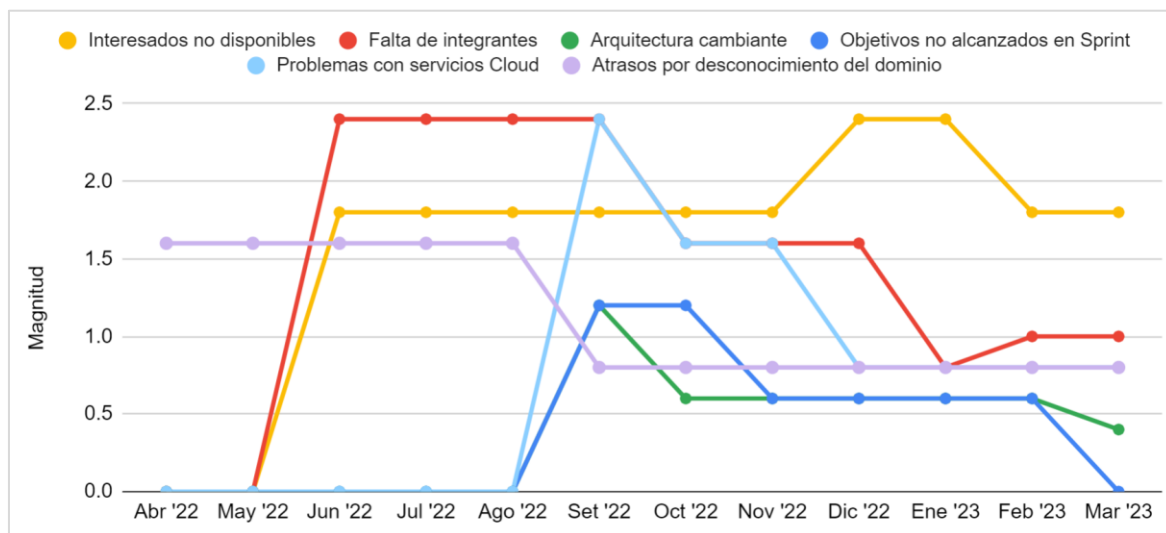


Figura 34 - Evolución de principales riesgos durante el proyecto

Vale aclarar que todos los riesgos ilustrados en la imagen se manifestaron, como se detalla a continuación.

- **Interesados no disponibles**

El riesgo de que los interesados no estuvieran disponibles para coordinar las validaciones se manifestó desde comienzos del proyecto, ya que el organizar reuniones con ellos no era fácil dado que se encontraban realizando obras y sus tiempos eran muy limitados, teniendo que cancelarlas y reagendar las instancias lo antes posible. Durante diciembre de 2022, la probabilidad de que los interesados no tuvieran disponibilidad aumentó dado que eran épocas festivas, siendo el mismo caso para enero de 2023, mes en donde tenía lugar el feriado de la construcción.

- **Falta de integrantes**

Por otra parte, uno de los riesgos de mayor magnitud fue la falta de un integrante, el cual se manifestó desde etapas iniciales del proyecto. En particular, durante la etapa de construcción, uno de nosotros estuvo enfermo durante una semana aproximadamente y, en otra ocasión, el otro tuvo imprevistos personales y no pudo dedicarles a sus tareas enteramente por unos días, por lo que el avance fue más lento. Sin embargo, en ambos casos, el integrante restante intentó ocuparse de las tareas que estaban asignadas al otro miembro, priorizándolas y pausando las suyas que no fueran de suma importancia. Este riesgo tuvo mayor magnitud entre junio y septiembre de 2022 puesto que durante este periodo nos encontrábamos en invierno, siendo más propensos a enfermarnos debido a las bajas temperaturas. A medida que nos acercábamos a la fecha de finalización del proyecto, la probabilidad de que faltara un integrante era poca, pero en caso de ser así hubiera tenido un impacto muy alto ya que nos encontrábamos realizando la documentación y el cierre del proyecto.

- **Problemas con servicios Cloud**

Otro de los riesgos de mayor magnitud, ocurrido en la etapa de construcción, fue el relacionado a los problemas con los servicios en la nube, ya que no teníamos permisos suficientes para poder configurar todos los servicios que requeríamos a través de los créditos ofrecidos por la Universidad ORT, lo que resultó en un desarrollo muy lento. En vista de esto y de que algunos servicios no estaban incluidos en el convenio, tomamos la decisión de crear una cuenta personal y postularnos a dos fondos brindados por AWS a estudiantes para utilizar sus servicios. Afortunadamente logramos conseguirlos, lo cual nos fue de gran ayuda para poder utilizar los servicios que necesitábamos más libremente y realizar pruebas de la plataforma.

- **Atrasos por desconocimiento del dominio**

Al igual que detallamos en la sección [2.5. Desafíos del proyecto](#), nosotros no contamos con conocimientos en el sector de la construcción y dependemos de nuestros interesados para validar que nuestra solución aporte valor al sector. Este riesgo se manifestó, en particular en donde tuvimos que reagendar reuniones con los interesados para poder validar nuestros resultados. Una vez comenzada la etapa de desarrollo, ya contábamos con más información para avanzar, aunque continuábamos

necesitando de validaciones para asegurarnos que nuestro producto aportaba valor al sector. Por este motivo, dicho riesgo disminuyó su magnitud, pero siguió manifestándose ya que tuvimos que volver a reagendar instancias. Es importante destacar que, pese a esto, logramos hacer las validaciones en los tiempos necesarios.

- **Arquitectura cambiante**

En el caso del riesgo relacionado a una arquitectura cambiante, el mismo se manifestó a partir de setiembre de 2022, correspondiente al inicio de la etapa construcción, donde hicimos modificaciones arquitectónicas para adaptar nuestra solución conforme íbamos obteniendo feedback de los interesados. Esto no quiere decir que la arquitectura estaba en constante cambio, sino que en las primeras iteraciones de la construcción notamos que era mejor tomar un enfoque distinto al planeado inicialmente, realizando pequeñas alteraciones. Además, dado que desde un principio tuvimos en cuenta ciertos patrones y principios de diseño, así como buenas prácticas tanto para el Frontend como el Backend, estos cambios tuvieron un bajo impacto en la estructura original, facilitando las posteriores implementaciones del resto de las funcionalidades.

- **Objetivos no alcanzados en Sprint**

Por último, el riesgo acerca de no poder alcanzar los objetivos en ciertos Sprints comenzó a manifestarse a partir de la etapa de construcción. En los primeros dos Sprints, surgieron imprevistos con servicios de AWS que estaban por fuera de lo definido, de manera que no pudimos cumplir con algunas de las otras tareas planificadas para dichas iteraciones, ocasionando que no pudiéramos incluir estos resultados en nuestras métricas (como se mencionó en la sección [8.6.2. Construcción](#)).

A su vez, queremos comentar otros dos riesgos que se manifestaron a pesar de que su magnitud no fue elevada, razón por la cual no los seleccionamos para mostrar en la gráfica.

- **Alcance del proyecto superado**

Este riesgo se manifestó en la primera etapa de gestión, la cual involucra la etapa de definición de la idea e investigación tecnológica, donde tuvimos que realizar una

priorización y reescritura de los RF, como se menciona en la sección [6.1.2. Priorización y reescritura](#).

- **Disputas entre integrantes**

Este riesgo está enfocado al equipo, ya que tuvimos instancias en donde hubieron malentendidos entre el equipo, pero aprendimos a organizarnos mejor a medida que fue avanzando el proyecto. Además, procuramos notificarnos más seguido acerca de nuestro estado y el de las tareas asignadas, así como incertidumbres o inconvenientes.

8.9. Conclusiones

A partir de lo que experimentamos, podemos concluir que la gestión de un proyecto es un proceso fundamental para lograr el éxito en su ejecución. La planificación adecuada es esencial para identificar los riesgos y oportunidades, establecer un plan anual, dividir las tareas a realizar y asignar los recursos necesarios. Del mismo modo, la identificación de los riesgos nos ayudó de gran manera a anticipar los posibles obstáculos y establecer medidas preventivas y correctivas en caso de ser necesario. La gestión de la comunicación también influyó fuertemente en que el equipo tuviera una comunicación fluida y transparente, tanto interna y con la tutora, así como con los interesados.

En resumidas cuentas, la gestión de un proyecto es un proceso complejo que requiere de una planificación rigurosa, el uso de herramientas y metodologías adecuadas, una gestión eficiente de los recursos, una documentación y un seguimiento constante. Teniendo en cuenta estos puntos, es que pudimos construir un proyecto que cumple con los objetivos planteados en un inicio.

9. Gestión de la configuración

Este capítulo contendrá las herramientas usadas para el manejo y control de versiones, incluyendo organización de los repositorios, gestión de cambios y plataformas utilizadas, tanto para el código fuente del producto en sí como para su documentación.

También se detallarán las herramientas de apoyo utilizadas para la construcción del código y documentación.

9.1. Herramientas utilizadas

En esta sección se detallarán las herramientas utilizadas para la gestión del código y la documentación, las cuales resultaron fundamentales para poder llevar adelante el proyecto de forma organizada y eficiente.

9.1.1. Código

Con el fin de mantener un registro de las distintas versiones del código y realizar un seguimiento de los cambios realizados, el código fuente fue almacenado utilizando GitHub como repositorio en la nube y Git para lo relativo al manejo de versiones. Esta decisión se debió a que esta plataforma se integra perfectamente con los servicios de AWS, permitiendo utilizar GitHub Actions para realizar chequeos pre y post merge y, en el caso de los servicios, implementar CI/CD.

En la siguiente imagen se puede visualizar como son ejecutadas estas acciones luego de mergear los cambios (Figura 35).

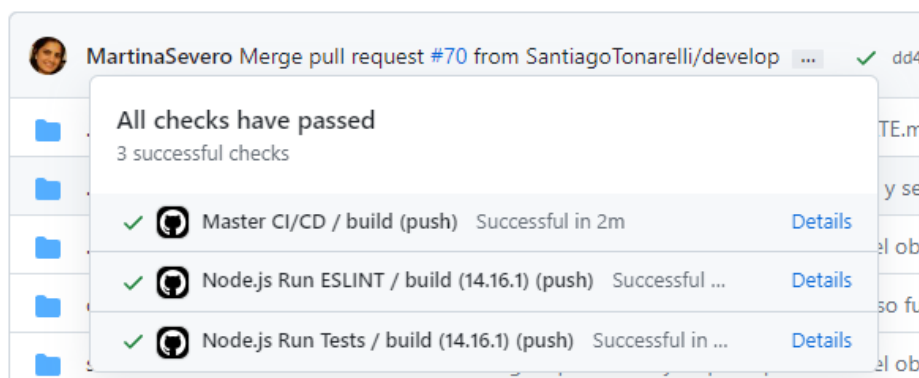


Figura 35 - Uso de GitHub Actions

En este caso al tratarse de los servicios, se corren los test, luego se valida que no haya errores en ESLint y finalmente se ejecuta la compilación y despliegue de servicios.

A su vez, GitHub nos permite el manejo de Pull Requests (PR) como se puede ver en la siguiente imagen (Figura 36), agregando una descripción de los cambios, comentarios y aprobación del reviewer.

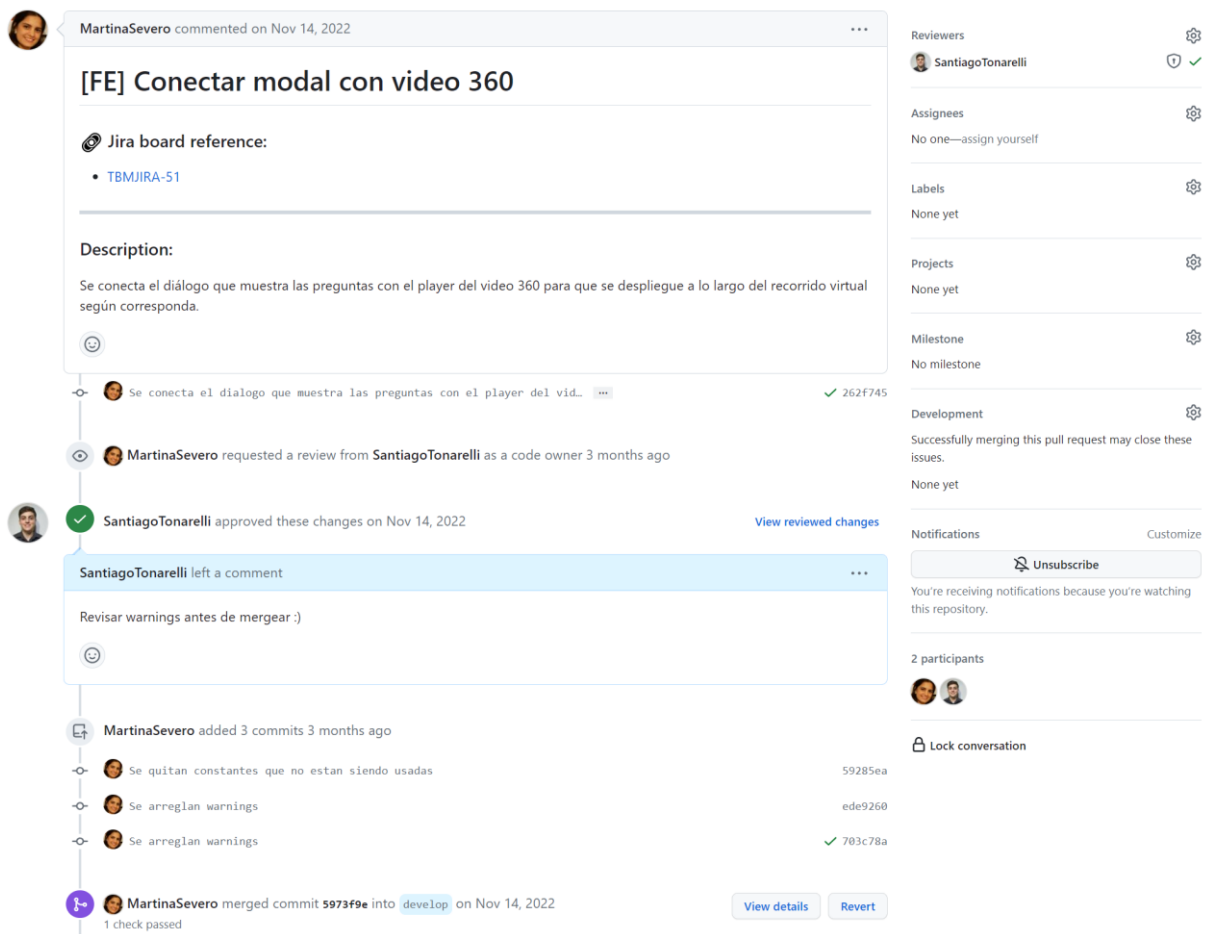


Figura 36 - Uso de Pull Request

9.1.2. Documentación

Para poder realizar un manejo efectivo de la documentación se utilizaron varias herramientas, las cuales detallaremos a continuación.

En primer lugar, para el diagramado se utilizó Miro. Esta plataforma nos fue muy útil ya que permite la creación de diagramas, tablas, notas y otros elementos para representar sucesos de forma visual y organizada (Figura 37).

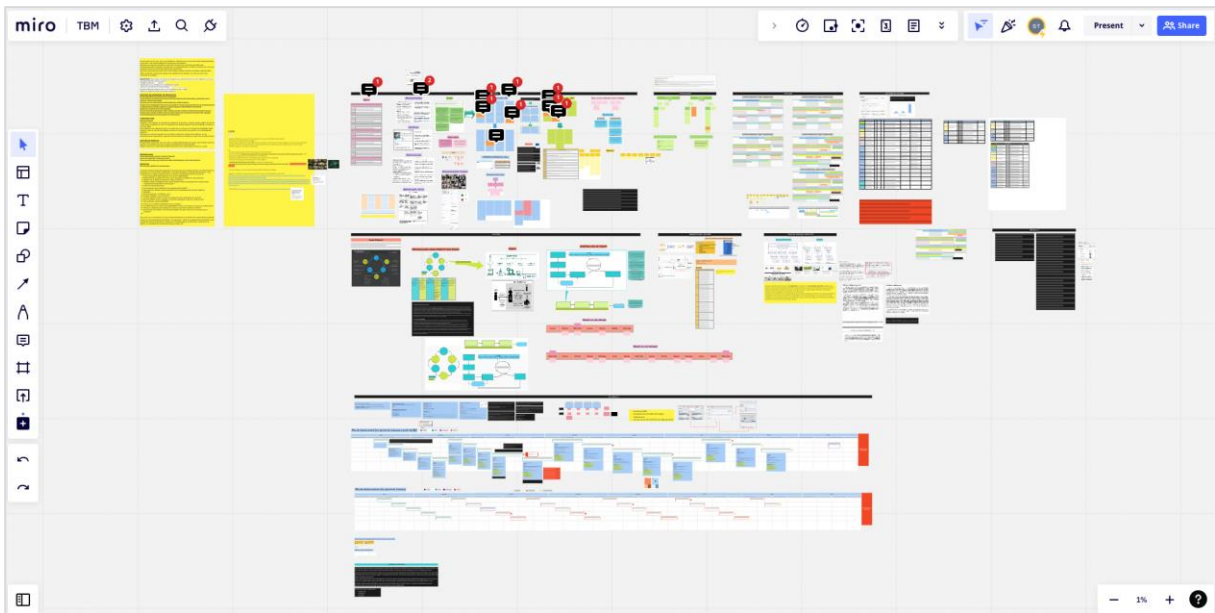


Figura 37 - Uso de Miro

A su vez se utilizó Google Drive para el almacenamiento de archivos importantes del proyecto, tales como diagramas (realizados en Draw.io), documentos y planillas. Esta herramienta fue de gran utilidad para mantener estos archivos organizados y accesibles desde cualquier lugar (Figura 38).

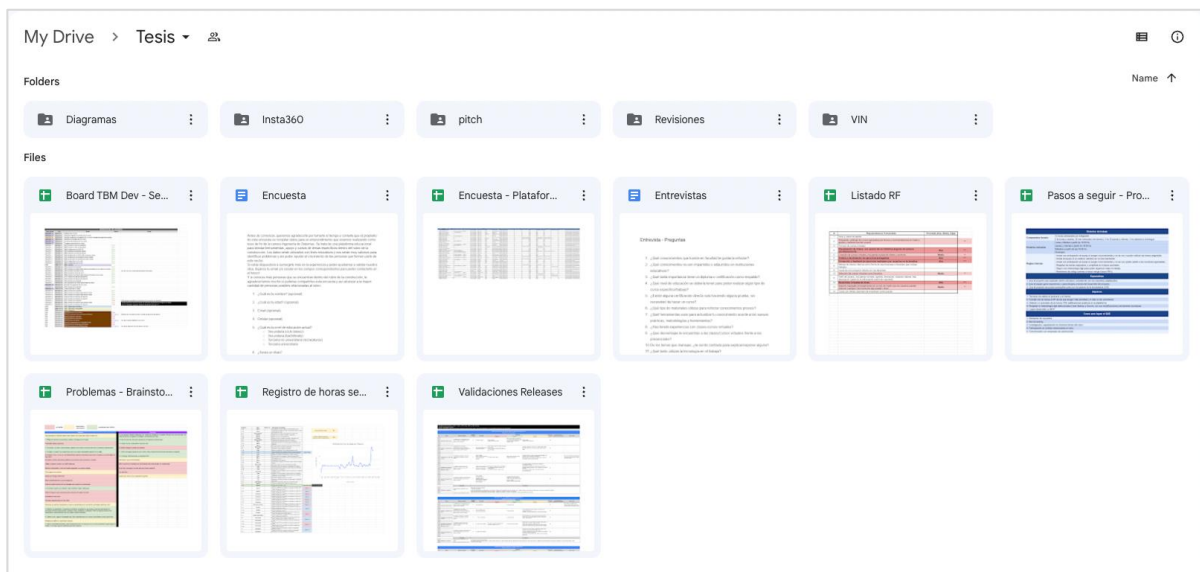


Figura 38 - Uso de Google Drive

Para la documentación a entregar se utilizó Microsoft Word y se respaldó en la nube proporcionada por OneDrive. Esto nos permitió el acceso remoto y seguro desde cualquier lugar, así como la colaboración en tiempo real para poder editar el documento simultáneamente, siendo una solución practica y efectiva para el manejo de la documentación del proyecto.

9.2. Organización de repositorios

Para la gestión de los repositorios, creamos dos para el proyecto. El primero es “TBM-Service”, el cual contiene todos los servicios relacionados a la plataforma y el segundo es “TBM-Webapp”, donde se encuentra almacenada la misma (Figura 39).

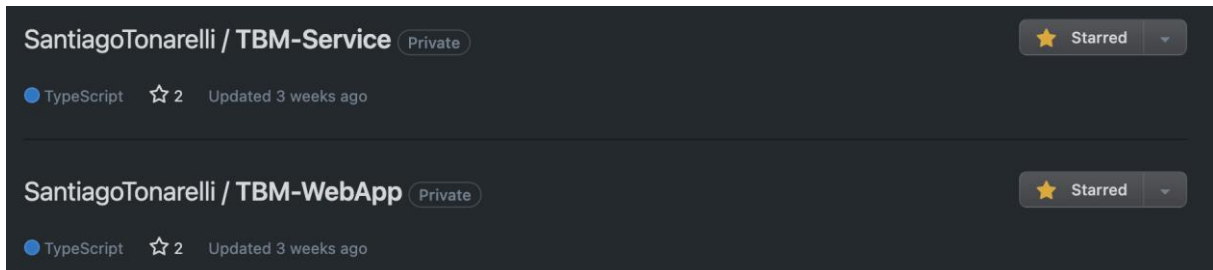


Figura 39 - Repositorios utilizados para el proyecto

Esta separación en dos repositorios nos permitió un manejo más eficiente y organizado de los distintos componentes del proyecto. Además, logramos un mayor control y seguimiento del código, ya que cada repositorio puede tener su propia estructura de ramas y versiones, facilitando la identificación de errores y la implementación de mejoras.

Como dato adicional, a continuación, mostramos la participación de ambos integrantes a través de las herramientas provistas por GitHub, tanto para el repositorio de Backend como el de Frontend (Figura 40 y Figura 41 respectivamente).

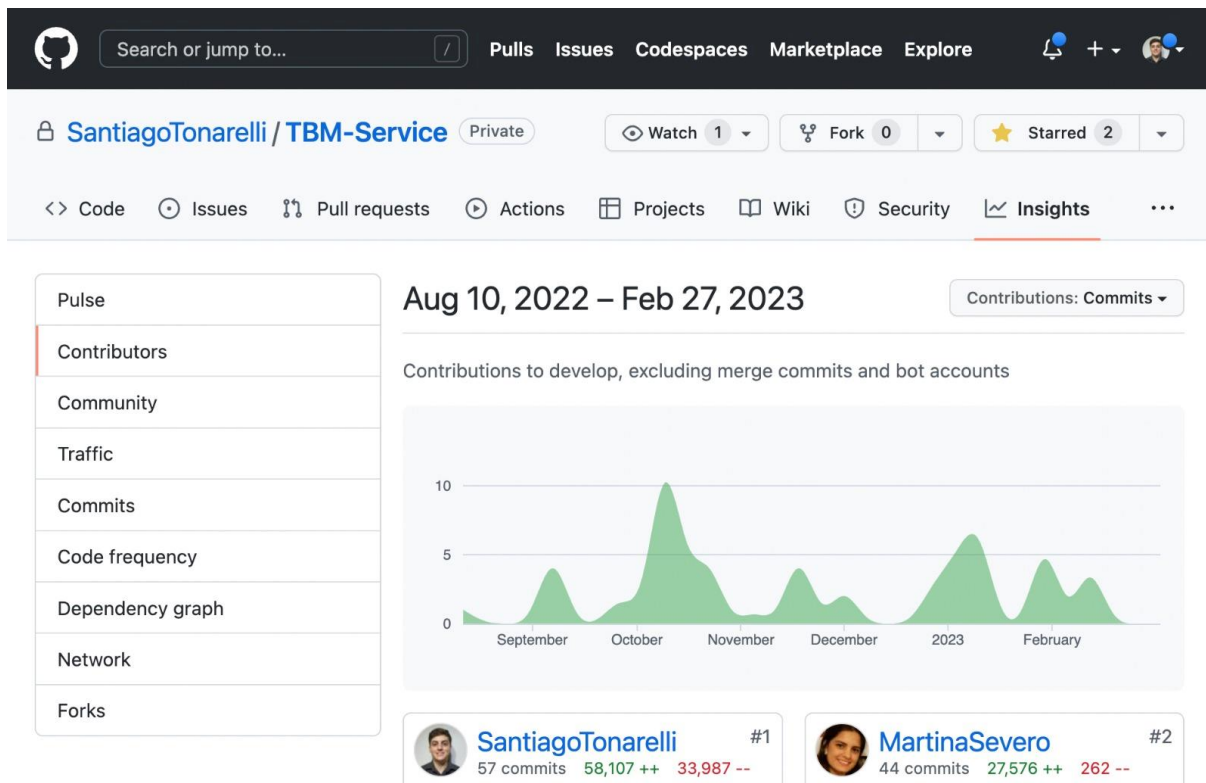


Figura 40 - Participación de los integrantes en el repositorio de Backend

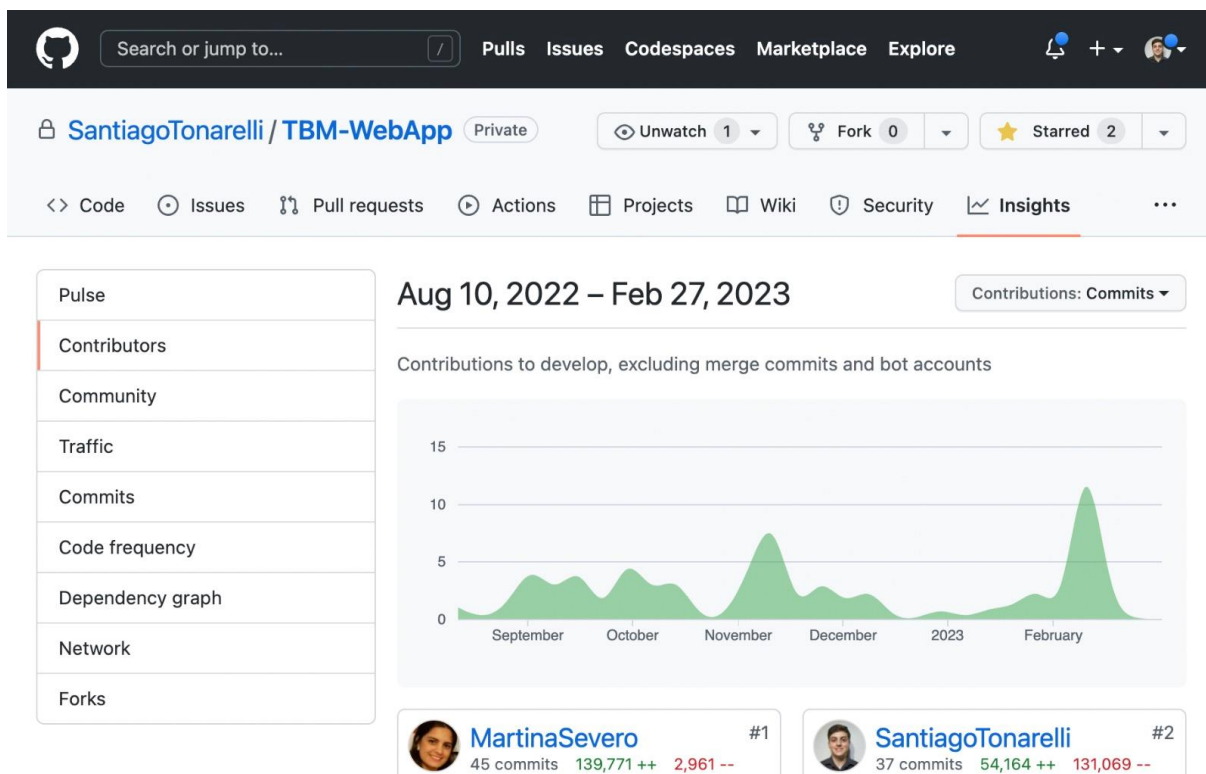


Figura 41 - Participación de los integrantes en el repositorio de Frontend

9.3. Control de versiones

El control de versiones fue un aspecto fundamental del proyecto, ya que nos permitió mantener un registro de los cambios realizados, tanto en el código como en la documentación, y tener un histórico de versiones para poder recuperar versiones antiguas si era necesario, además de facilitar la colaboración entre los integrantes del equipo.

En esta sección, abordaremos el control de versiones del código y la documentación del proyecto, utilizando las respectivas herramientas, mencionadas en la sección [9.1. Herramientas utilizadas](#).

9.3.1. Código

Para llevar a cabo el control de versiones del código, utilizamos lo establecido por el modelo de ramificación para Git conocido como GitFlow. [31]

En cuanto a la estructura de los repositorios, en primer lugar, creamos una rama llamada *develop*, a partir de la rama principal *master*. Es importante destacar que no trabajamos directamente sobre ninguna de estas ramas. En caso de corregir algún bug luego de un Release, abrimos una rama desde *master* (HotFix), se solucionará el problema y se volverá a “mergear” con *master*. Luego, por cada tarea contenida dentro de las User Story, abrimos una rama *feature* específica para el desarrollo de esa funcionalidad. Una vez implementada la misma, mergeamos esa nueva rama a *develop*. En el caso de la finalización de un Sprint que tiene como resultado un Release, a diferencia de GitFlow, no se crea una rama para el Release, sino que se realiza un único merge de *develop* a la rama principal *master*.

A modo de comentario, en el caso particular de los bugs, los cuales no figuran en el modelo GitFlow, simplemente utilizamos la notación “bugfix” al crear una rama orientada al arreglo del bug para distinguirla de las features.

Es importante destacar que, en muchos casos se realizaron varias tareas que estaban relacionadas en la misma rama, debido a que se buscó optimizar el tiempo de revisión de código. En estos casos, se detalló en las Pull Request.

A continuación, se puede apreciar el uso de las ramificaciones mencionadas de GitFlow (Figura 42).

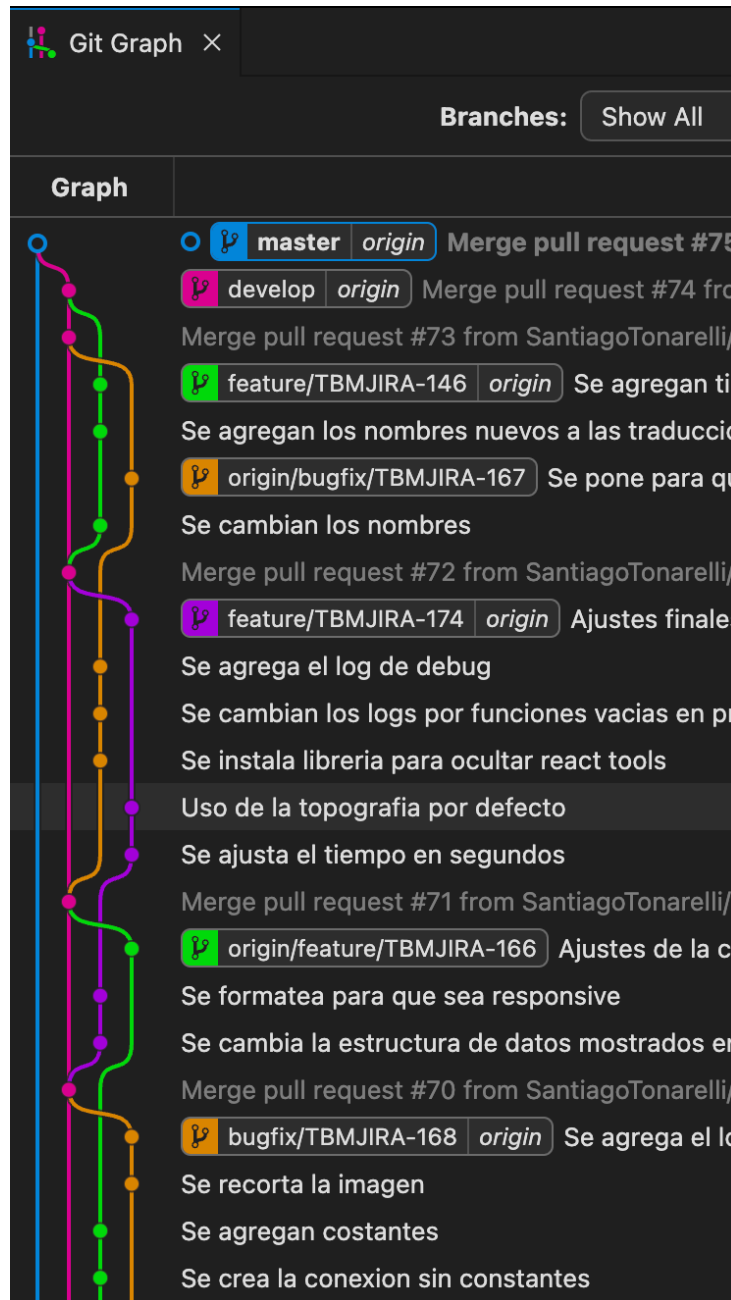


Figura 42 - Uso de GitFlow para control de versiones del código

A continuación, mostramos también la evidencia de los Releases realizados con las pruebas de las GitHub Actions, incluido el correcto deploy, para los repositorios de Backend y Frontend (Figura 43 y Figura 44 respectivamente).

SantiagoTonarelli / TBM-Service Private

Watch 1 Fork 0 Starred 2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Filters is:pr Develop - Master [Release] Labels 9 Milestones 0 New pull request

Clear current search query, filters, and sorts

<input type="checkbox"/>	0 Open	6 Closed	Author	Label	Projects	Milestones	Reviews	Assignee	Sort
<input type="checkbox"/>			Develop - Master [Release 6] ✓						1
			#70 by MartinaSevero was merged last week • Approved						
<input type="checkbox"/>			Develop - Master [Release 5] ✓						
			#56 by MartinaSevero was merged 3 weeks ago • Approved						
<input type="checkbox"/>			Develop - Master [Release 4] ✓						1
			#44 by MartinaSevero was merged on Jan 25, 2023 • Approved						
<input type="checkbox"/>			Develop - Master [Release 3] ✓						
			#37 by SantiagoTonarelli was merged on Dec 28, 2022 • Approved						
<input type="checkbox"/>			Develop - Master [Release 2] ✓						
			#24 by MartinaSevero was merged on Nov 30, 2022 • Approved						
<input type="checkbox"/>			Develop - Master [Release 1] ✓						
			#15 by SantiagoTonarelli was merged on Oct 19, 2022 • Approved						

Figura 43 - Releases realizados correspondientes al repositorio de Backend

SantiagoTonarelli / TBM-WebApp Private

Unwatch 1 Fork 0 Starred 2

Code Issues Pull requests Actions Projects Wiki Security 4 Insights Settings

Filters is:pr Develop - Master [Release] Labels 9 Milestones 0 New pull request

Clear current search query, filters, and sorts

<input type="checkbox"/>	0 Open	6 Closed	Author	Label	Projects	Milestones	Reviews	Assignee	Sort
<input type="checkbox"/>			Develop - Master [Release 6] ✓						1
			#75 by MartinaSevero was merged last week • Approved						
<input type="checkbox"/>			Develop - Master [Release 5] ✓						
			#60 by MartinaSevero was merged 3 weeks ago • Approved						
<input type="checkbox"/>			Develop - Master [Release 4] ✓						1
			#54 by MartinaSevero was merged on Jan 25, 2023 • Approved						
<input type="checkbox"/>			Develop - Master [Release 3] ✓						1
			#39 by MartinaSevero was merged on Dec 29, 2022 • Approved						
<input type="checkbox"/>			Develop - Master [Release 2] ✓						
			#28 by MartinaSevero was merged on Nov 30, 2022 • Approved						
<input type="checkbox"/>			Develop - Master [Release 1] ✓						1
			#17 by MartinaSevero was merged on Oct 20, 2022 • Approved						

Figura 44 - Releases realizados correspondientes al repositorio de Frontend

9.3.2. Documentación

En cuanto a la documentación, utilizamos la herramienta Word para desarrollarla, junto con OneDrive. Ambas nos permitieron mantener un control de versiones de dicha documentación. Sin embargo, fuimos realizando copias cada dos días puesto que, en la versión de escritorio para Mac, al editar el mismo archivo de forma simultánea, este no se sincronizaba correctamente o corrompía la versión original. Por este motivo, decidimos hacer un versionado manual que nos permitió recuperar el progreso del documento en reiteradas ocasiones.

9.4. Conclusiones

La correcta gestión de la configuración influyó de gran manera para que pudiéramos desarrollar un producto de calidad a la vez que trabajamos en paralelo en distintas tareas. Al mantener un control de versiones adecuado y utilizar herramientas de software especializadas, pudimos avanzar de manera más eficiente y coordinada. El uso de un sistema de control de versiones nos permitió realizar un seguimiento exhaustivo de los cambios realizados en nuestra plataforma, lo que facilitó la identificación de errores y la resolución de problemas.

Como aprendizaje, en cuanto a las herramientas que usamos para la documentación, concluimos que es necesario manejar un control de versiones manual cada cierto tiempo, dado que muchas de estas herramientas ofrecen mecanismos de guardado automático y versionado, pero que en el caso de corromperse algo, es muy difícil, en base a nuestra experiencia en este proyecto, seguir la trazabilidad de un cambio. Por otro lado, esto si es posible con las ramificaciones que utiliza Git.

10. Gestión de calidad

Este capítulo estará orientado a detallar que es un “producto de calidad” para nosotros, las actividades de QA realizadas, incluyendo tests unitarios, validación de criterios de aceptación, tanto de los RF como RNF y la validación con los interesados. Además, mencionaremos la importancia de realizar una correcta definición de las User Stories y las métricas que validen el cumplimiento de la gestión de la calidad. También comentaremos los bugs encontrados en estos procesos y si persiste alguno conocido en nuestra solución final.

10.1. Definición de calidad para TechBuildMe

Para nosotros, la calidad representa, en primer lugar, que nuestra plataforma cumpla con los estándares de calidad establecidos en la industria, así como con atributos de calidad como la Usabilidad, Modificabilidad, Disponibilidad, Eficiencia, Seguridad y Portabilidad, como se menciona en la sección [1.3.3. Objetivos del proyecto](#).

En segundo lugar, que nuestros interesados estén satisfechos con el producto y que se sientan motivados a usarlo y a su vez inviten a otros a utilizarlo. Por eso consideramos que realizar validaciones de prototipos, RF y Releases para obtener su feedback y finalmente, del MVP, hacen que nuestro producto se acerque lo más posible a cumplir con las necesidades que identificamos que existen en el mercado, a través del relevamiento de datos y análisis que hicimos inicialmente (para más detalle ver [4. Ideación inicial](#)).

Asimismo, como mencionamos a lo largo de este documento, los RF implementados son la base para hacer que la experiencia de usuario sea la más completa en cuanto a lo que es capacitación online dentro del rubro de la construcción. Lo relativo a los cursos es fundamental para el aprendizaje, sin ello directamente no podemos ofrecer capacitaciones a través de cursos virtuales.

Los recorridos virtuales forman la parte más compleja de nuestra plataforma y representan la propuesta de valor; ofrecer la posibilidad de realizar un tour virtual por una obra desde cualquier lugar, sin tener que trasladarse físicamente, permitiendo al usuario hacer un uso más eficiente de su tiempo.

En cuanto al feedback y ejercicios entregables, son una forma en que los estudiantes pueden poner en práctica lo aprendido y también recibir comentarios acerca de su progreso, lo cual resulta en una mejor experiencia de aprendizaje.

10.2. Prácticas de aseguramiento de la calidad

A continuación, mencionaremos las prácticas que seguimos para poder aportar a una mayor calidad de nuestro producto.

10.2.1. Uso de estándares

En primer lugar, detallaremos el uso de ciertos estándares para los distintos aspectos del proyecto, con el fin de que el proyecto tuviera uniformidad y fuera mantenible en el tiempo.

▪ Código

A nivel de código, se siguieron los estándares que detallaremos a continuación.

○ Clean Code

Nos pareció importante tener una forma de asegurarnos que fuéramos consistentes al momento de programar, tanto en Frontend como en Backend. Por este motivo, decidimos emplear prácticas dictadas por Clean Code adecuado al estándar del lenguaje utilizado (en nuestro caso TypeScript) las cuales mencionamos a continuación:

- Escritura de nombres de variables, funciones, métodos y constantes en lowerCamelCase.
- Uso de nombres sin ambigüedad, que revelen su intención, para variables, funciones, métodos y constantes.
- Creación de funciones y métodos que no superen las 30 líneas de código.
- Creación de funciones que no reciban más de 3 argumentos.
- Evitación de código duplicado.

- Remoción de variables, funciones, métodos, constantes e imports sin usar.
- Evitación de comentarios innecesarios.

En los siguientes puntos mencionaremos las herramientas que nos ayudaron a cumplir con Clean Code. Asimismo, las revisiones de código realizadas en las Pull Request nos permitieron identificar algunos aspectos que escapan de las herramientas, sirviendo así como una segunda barrera para detectar estos incumplimientos.

- **Extensiones**

A modo de ayuda y seguimiento de estándares de la industria, utilizamos las extensiones ESLint, para ayudarnos a arreglar problemas que pueda haber en nuestro código y Prettier, para formatear el código, en Visual Studio Code.

En el caso del Frontend, el correr el proyecto y que no se encuentren issues, es la forma de evidenciar que no hubo errores ni warnings en el código (Figura 45).

```
Compiled successfully!

You can now view tbm-webapp in the browser.

  Local:            http://localhost:3000
  On Your Network: http://192.168.1.65:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
No issues found.
□
```

Figura 45 - Evidencia de uso de ESLint (Frontend)

Con respecto a nuestras funciones, al correr el comando de ESLint y no obtener ninguna salida, significa que nuestro código no tiene errores ni warnings (Figura 46).

```
● (base) → TBM-Service git:(master) ✖ npm run lint
> tbm-service@1.0.0 lint /Users/santiagotonarelli/Desktop/TBM/TBM-Service
> eslint --ext .js,.ts .
```

Figura 46 - Evidencia de uso de ESLint (Backend)

- **Codificación en inglés**

Debido a que el contenido y fuentes de código en nuestra industria está mayormente en este idioma, decidimos escribir el código en inglés. Para asegurarnos de no tener faltas de ortografía, usamos la extensión Code Spell Checker de Visual Studio Code.

- **User Stories**

Para el caso de las historias de usuario, dispusimos una serie de condiciones necesarias para considerarlas como completas (Definition of Done):

- Cumplimientos de los correspondientes criterios de aceptación definidos.
- Codificación con los estándares anteriormente detallados.
- Creación de un commit y subida de la rama correspondiente a la tarea. En el caso del Frontend, Husky chequea que no haya errores relacionados a ESLint. En el caso del Backend que no haya errores de ESLint y que los tests unitarios pasen completamente (en caso contrario no permite subir la rama al repositorio).
- Creación de Pull Request (PR) con el formato correspondiente (para más detalle ver [13.18. Anexo 18: Template de Pull Request](#)).
- Chequeo de GitHub Actions exitoso.
- Aprobación de la PR por el otro integrante y merge de la rama por el creador.
- Actualización del estado de la tarea correspondiente en el tablero de Jira a Done.
- Correcto funcionamiento del build y deploy. En el caso del Frontend, se utilizó AWS Amplify Hosting. En el caso del Backend, se hizo el build y deploy del archivo serverless mediante GitHub Actions).

10.2.2. Revisiones

A continuación, mencionaremos las prácticas que llevamos a cabo enfocadas al seguimiento del estado de nuestro proyecto.

- **Pull Requests**

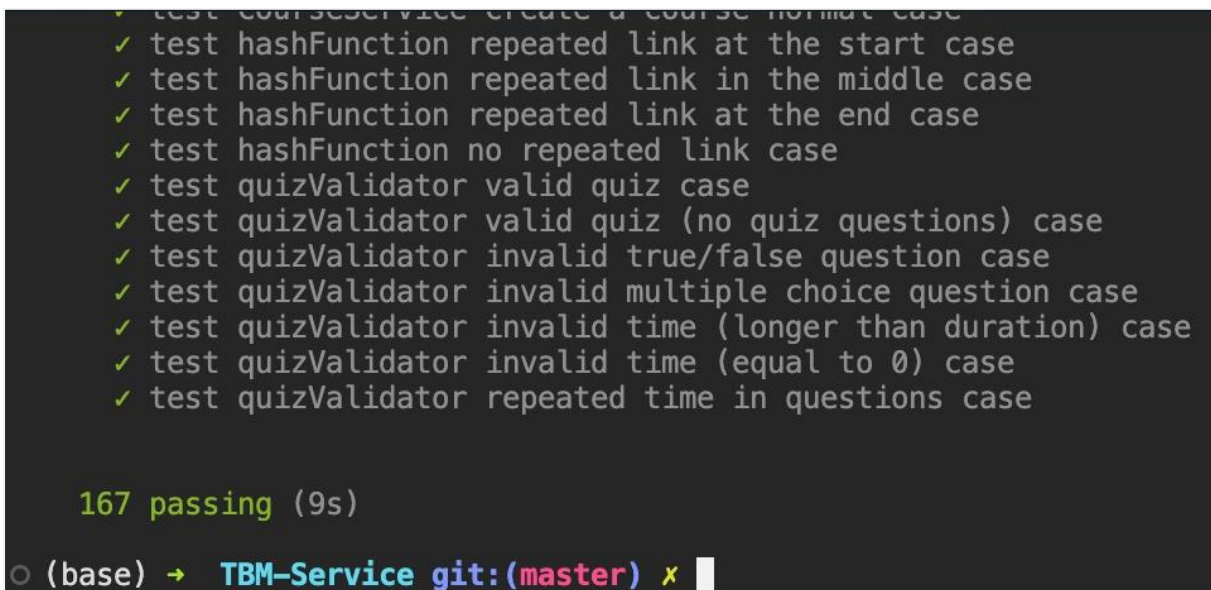
El agregar al otro integrante como revisor al crear una PR cuando se terminaba una tarea ayudó a mantenernos al tanto de los cambios realizados por el otro y hacer correcciones, si es que detectábamos alguna.

- **Revisiones por parte de la Universidad ORT**

A lo largo del proyecto tuvimos tres instancias donde recibimos feedback de revisores, lo cual nos fue muy útil para conocer sus puntos de vista y recibir sugerencias para posibles mejoras que podíamos realizar, o agregar información faltante.

10.2.3. Tests unitarios

Para nuestro Backend, realizamos un total de 167 pruebas unitarias a lo largo de la etapa de construcción, asegurándonos que lo que representamos allí funciona correctamente (Figura 47).



```
✓ test courseService create a course normal case
✓ test hashFunction repeated link at the start case
✓ test hashFunction repeated link in the middle case
✓ test hashFunction repeated link at the end case
✓ test hashFunction no repeated link case
✓ test quizValidator valid quiz case
✓ test quizValidator valid quiz (no quiz questions) case
✓ test quizValidator invalid true/false question case
✓ test quizValidator invalid multiple choice question case
✓ test quizValidator invalid time (longer than duration) case
✓ test quizValidator invalid time (equal to 0) case
✓ test quizValidator repeated time in questions case

167 passing (9s)

○ (base) → TBM-Service git:(master) x █
```

Figura 47 – Resultado de los tests unitarios

Pese a que las pruebas unitarias tuvieron como fin únicamente probar la lógica de negocio, y no las implementaciones completas con la base de datos, es decir, como se muestra en la vista de módulos en la sección 7.6.1. Vista de módulos, en particular, en la vista de módulos Layers, las únicas capas que se chequearon fueron Controllers y Services, obtuvimos resultados de cobertura superiores al 85% como se puede ver en la Figura 48.

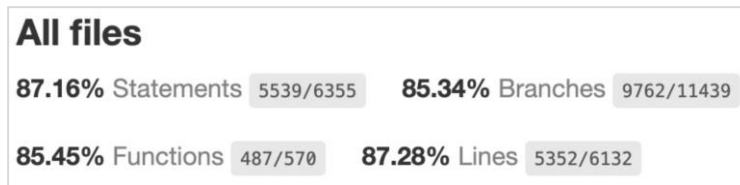


Figura 48 – Resultados de la cobertura

Esto nos permitió asegurar que nuestras pruebas unitarias garantizan la funcionalidad de nuestro Backend.

Dichas pruebas unitarias siguen el formato ATRIP. A su vez, cada prueba tiene la estructura AAA (Arrange, Act, Assert) y, para garantizar su unicidad, se usaron mocks dentro de las pruebas (Figura 49).

```

it('test UserService create a user', async () => {
  // Arrange
  const userExpected = new User('test-andy@email.com', 'andy@email.com', 'test', 'Andrew', 'Law',
    'andrew-picture');

  const product = {
    productId: 'test',
    data: {}
  };

  const mockedUserRepository = mock<IUserRepository>();
  when(mockedUserRepository.createUser(anyOfClass(User), anything())).thenResolve(userExpected);
  when(mockedUserRepository.addNotification(anyString(), anything(), anything(), anything()));

  const courseService = new UserService(instance(mockedUserRepository));

  // Act
  const result = await courseService.createUser(userExpected, product);

  // Assert
  verify(mockedUserRepository.createUser(anyOfClass(User), anything())).once();
  verify(mockedUserRepository.addNotification(anyString(), anything(), anything(), anything())).once();
  expect(result).to.deep.equalInAnyOrder(userExpected);
});

```

Figura 49 - Ejemplo de prueba unitaria realizada (Backend)

En el caso del Frontend, verificamos que se cumplieran los criterios de aceptación con los flujos establecidos, además de los bugs solucionados y mejoras obtenidas de las validaciones.

10.3. Validaciones

En cuanto a las validaciones con los interesados, tuvimos varias instancias, las cuales detallaremos a continuación, a partir de las cuales dejamos registrados los resultados en las herramientas utilizadas para ello, como Excel o Miro. Queremos destacar que

ambos estuvimos presentes en todas las instancias, de modo que uno explicara e hiciera las preguntas a los interesados y el otro se encargara de tomar notas y asegurarse de que no se pasaran cosas por alto.

10.3.1. Prototipos

Para los prototipos, la modalidad de la reunión con cada interesado se basó en, a través de Zoom, compartirles pantalla e ir explicándoles los distintos prototipos que elaboramos (para más detalle ver [13.4. Anexo 4: Prototipos realizados](#)). Les preguntábamos su opinión al respecto y si tenían mejoras o si notaban algo que no fuera de su agrado.

En esta primera instancia, contamos con el apoyo de todos los interesados mencionados en la sección [2.4. Interesados que nos acompañaron](#), donde recolectamos varios comentarios positivos (Figura 50).

Reunion Camila Diaz 11-07-22	Reunion Fernanda Ferrari 24-06-22	Reunion Diego Perez 05-07-22	Reunion Empresa Stiler 18-06-22
Mencionó que también estaría bueno pedir nombre y apellido en la pantalla de registro.	Le gustó la simplicidad de la UI.	Le gustó que la UI fuera sencilla y fácil de entender.	Les gustó la idea en general.
Manifestó aprobación por los RF de la plataforma.	Para los videos con varios ángulos de cámara, no le agradó mucho que aparezca un diálogo para cambiar las posiciones.	Le gustó que se pudiera agregar calificación y favoritos desde la página de visualización detallada de un curso.	Nos dijeron que estaban buscando algo similar con realidad virtual (empresa europea).
Hizo enfoque en los recorridos virtuales y actividades dinámicas.	Le pareció buena idea lo de los recorridos virtuales.	Para el recorrido virtual, comentó que las instrucciones podrían mostrarse en la página inicial de los mismos en vez de estar fijo cuando se ingresa a uno en particular. O podría esconderse.	Les gusto la idea de distintos angulos y los recorridos virtuales.
Le pareció bueno el recibir feedback de los docentes.		Para el login y signín, le parecieron bien google y facebook como aplicaciones más utilizadas. Mencionó que él también ingresa a otros sitios con su cuenta de Apple.	Mencionan la idea de tener una plataforma interactiva.
		Manifestó aprobación por los RF de la plataforma.	Destacaron la importancia de la simpleza.
		Resaltó que ciertas personas en el rubro pueden tener bajos niveles de educación.	Hizo mucho énfasis en hacer cursos cortos o que estos esten dividíos en modulos.

Figura 50 - Feedback de la validación de prototipos

Muchos de los comentarios positivos acerca de ese primer acercamiento a nuestra aplicación estuvieron orientados a estar de acuerdo con la sencillez y facilidad de entendimiento de la interfaz de usuario.

10.3.2. Requerimientos funcionales

Luego, pasamos a definir los RF y volvimos a reunirnos con los interesados para validarlos y priorizarlos, donde registramos las opiniones que íbamos recolectando en una planilla Excel (para más detalle ver [6.1.2. Priorización y reescritura](#)).

10.3.3. Releases

Para validar los Releases que hicimos, coordinamos instancias recurrentes con los interesados mediante Zoom. Primero, les explicábamos cual iba a ser la dinámica que íbamos a realizar con ellos para asegurarnos que no tuvieran ningún inconveniente, les pasábamos el enlace de producción de nuestra aplicación y les pedíamos que compartieran pantalla. Luego, según el RF que correspondiera, les pedíamos realizar una serie de pasos, los cuales elaboramos antes de cada instancia de validación sin describirles cómo usar la aplicación para lograr cumplirla (Figura 51). El objetivo de esta actividad era observar sus reacciones al interactuar con ella y si lograban realizar la acción solicitada por sí mismos, de modo que evaluáramos que tan usable e intuitivo era nuestro producto.

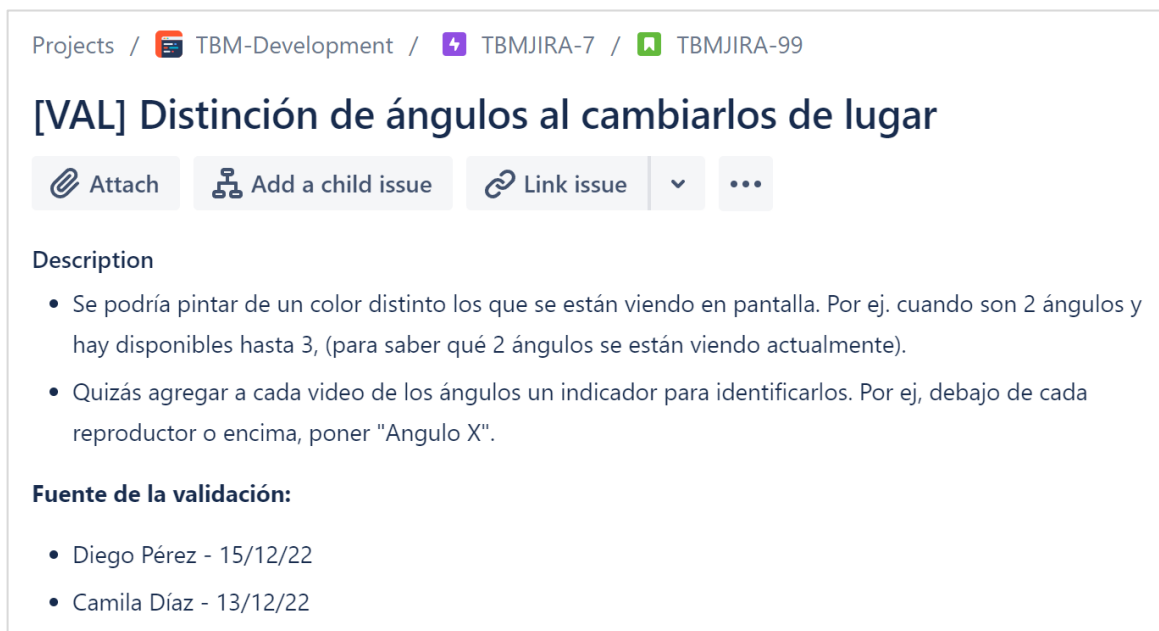
#	Paso	Acción a realizar
1	Vista previa de un curso.	Si quisieras ver la vista previa de un curso, es decir, sus detalles a gran escala, cómo lo harías?
2	División en módulos de un curso.	Si quisieras ver como está conformado un curso, es decir, si tiene una lista de temas, secciones o algo del estilo, cómo lo harías?
3	Visualización de videos de un curso.	Si quisieras visualizar un video de una sección en particular, cómo lo harías?
4	Visualización de distintos ángulos de cámara de video de un curso.	Si quisieras visualizar más de un ángulo de cámara para un video, cómo lo harías?
5	Sección de preguntas de un curso.	Si quisieras visualizar comentarios de otros usuarios acerca de un curso, cómo lo harías?

Figura 51 - Ejemplo de serie de pasos de una validación de un Release

Una vez que completaban la acción, les pedíamos que puntuaran del 1 a 5 que tan usable consideraban que era y por qué. En caso contrario los deteníamos para explicarles como lo hubieran conseguido. Posteriormente les solicitábamos comentarios acerca de lo que habían visto al realizar el flujo, tanto positivos como negativos y posibles mejoras que pudieran sugerirnos para mejorar la experiencia del usuario. Finalmente, preguntábamos acerca de sus opiniones de la aplicación en general y si consideraban que aportaba valor al sector y de qué forma.

En total tuvimos seis instancias de validación, cada una correspondiente a un Release. En cuanto a la usabilidad de nuestra solución final, obtuvimos un promedio de 4,5 de 5, siendo este un valor muy satisfactorio. Los resultados obtenidos en dichas instancias pueden ver en [13.20. Anexo 20: Feedback de Releases](#).

Una vez obtenido el feedback, procedimos a efectuar el proceso de aprendizaje, detallado en la sección, [5.3.2. Aplicación de las metodologías](#), realizando un análisis de estos resultados para así transformarlos en tareas que agregamos a nuestro Backlog. Dichas tareas contenían una descripción con el comentario recibido, que interesado/s lo mencionaron y en que instancia de validación. Se puede ver un ejemplo de esto en la Figura 52.



The screenshot shows a Jira issue page. At the top, the breadcrumb navigation reads: 'Projects / TBM-Development / TBMJIRA-7 / TBMJIRA-99'. The issue title is '[VAL] Distinción de ángulos al cambiarlos de lugar'. Below the title are several action buttons: 'Attach', 'Add a child issue', 'Link issue', a dropdown arrow, and a three-dot menu. The 'Description' section contains two bullet points: 'Se podría pintar de un color distinto los que se están viendo en pantalla. Por ej. cuando son 2 ángulos y hay disponibles hasta 3, (para saber qué 2 ángulos se están viendo actualmente).', and 'Quizás agregar a cada video de los ángulos un indicador para identificarlos. Por ej, debajo de cada reproductor o encima, poner "Angulo X".'. Below the description is the 'Fuente de la validación:' section, which lists two items: 'Diego Pérez - 15/12/22' and 'Camila Díaz - 13/12/22'.

Figura 52 - Tarea correspondiente a una validación de Release 2

En total, creamos 21 tareas a partir de las validaciones, pudiendo implementar 12, donde el resto no pudieron ser llevadas a cabo debido a que no contamos con el tiempo suficiente para desarrollarlas.

Al inicio de cada Sprint, en la ceremonia de Sprint Planning, evaluamos que tareas correspondientes a las validaciones podían ser implementadas en el próximo Sprint, en base al tiempo disponible, la complejidad de la tarea de la validación y las demás tareas que había para realizar. A su vez, teniendo en cuenta lo anterior, les asignamos prioridades, pudiendo tomar los valores Alta, Media o Baja.

A continuación, mencionamos las tareas creadas y realizadas a partir de los resultados de las validaciones y que fueron completadas durante el proyecto (Tabla 23). Es importante destacar que todas las validaciones de prioridad Alta fueron realizadas. Para ver más en detalle todas las tareas, ir a [13.21. Anexo 21: Tareas de las validaciones](#).

TBMJIRA-7 Validaciones		
ID	Nombre	Prioridad
TBMJIRA-65	[VAL] Agregar distintas velocidades de reproducción	Alta
TBMJIRA-66	[VAL] Poder intercambiar ángulos de cámara	Alta
TBMJIRA-67	[VAL] Cambiar tamaño de reproductores de video	Alta
TBMJIRA-68	[VAL] Agregar punteado breve del contenido del curso en "Acerca de"	Media
TBMJIRA-98	[VAL] Poder deshabilitar/habilitar quiz en recorrido virtual	Alta
TBMJIRA-99	[VAL] Distinción de ángulos al cambiarlos de lugar	Alta
TBMJIRA-100	[VAL] Reproductor de video en segundo plano	Media
TBMJIRA-101	[VAL] Agregar breve info para distinguir los recorridos virtuales	Media
TBMJIRA-124	[VAL] Agregar formato y tamaño de entrega	Media
TBMJIRA-125	[VAL] Mensaje informativo para registro	Alta
TBMJIRA-144	[VAL] Vista de docente y estudiante no está clara	Media
TBMJIRA-145	[VAL] Notificaciones	Media
TBMJIRA-146	[VAL] Términos referentes a Ejercicios generan confusión	Media
TBMJIRA-173	[VAL] Cambiar tonos de ciertos colores	Media
TBMJIRA-174	[VAL] Cambiar datos mostrados en listado de cursos	Media

Tabla 23 - Tareas creadas a partir de las validaciones

10.3.4. MVP

La validación del MVP constó en obtener la satisfacción de los interesados con respecto a la plataforma web lograda con el Release 4. En este caso, la usabilidad promedio tuvo un valor de 4,4 de 5, lo cual consideramos un resultado muy positivo.

Finalizada esta validación en particular, les pedimos que nos compartieran sus opiniones de manera más concreta acerca de lo que habían visto hasta el momento y el valor que creían que aportaba al sector. A continuación, detallamos los comentarios obtenidos de parte de cada uno de los interesados.

- **Fernanda (Arquitecta)**

"Como arquitecta con experiencia, debo decir que esta plataforma se destaca por su propuesta de valor basada en videos de alta calidad desde diferentes ángulos de cámara, videos de realidad virtual con preguntas en el recorrido y ejercicios con feedback específicos. Considero que los videos de diferentes ángulos de cámara permiten comprender mejor los procesos que se muestran y los videos de realidad virtual me parecen muy útiles para experimentar el entorno de una obra en tiempo real. Los ejercicios y el feedback están buenos para lo que es poder practicar lo que se va viendo en los cursos. Creo que esta plataforma es una herramienta muy valiosa para el sector de la construcción que puede ayudar a los profesionales a mejorar sus habilidades y adquirir nuevos conocimientos de manera más eficiente."

- **Camila (Arquitecta)**

"Una de las cosas que más me gustó fue la posibilidad de ver videos con diferentes ángulos de cámara. También lo de tener recorridos virtuales en obras me parece super interesante y que te aparezcan preguntas también me parece que está bueno para repasar conceptos. A pesar de que me gradué no hace mucho tiempo, en mi opinión creo que la plataforma le puede servir mucho al sector y que puede aportar valor a cualquier persona interesada en la construcción."

- **Diego (Ingeniero civil)**

"Creo que hoy en día hay muchas plataformas que se dedican a la enseñanza online pero no he visto, o por lo menos ahora no recuerdo ninguna que fuera dedicada cien por ciento al rubro de la construcción. Pienso que esta plataforma está buena en lo relacionado a temas de calidad, como lo de tener varios ángulos de cámara para los videos y los recorridos 360 en obras. Me parece que está bueno para probar en un principio, sobre todo creo que les va a resultar interesante a personas que inician en el rubro. Eso de tener idas y vueltas con el tema de los ejercicios y el feedback de un docente también aporta. En ingeniería hay muchas formas de encarar los problemas y puede que no lo veas como el docente lo propone pero que tampoco sea incorrecto y en eso ambas partes pueden complementarse. Me parece que puede ser una muy buena plataforma de aprendizaje que puede mostrar muchos enfoques, con el tema

de varios ángulos de cámara, videos 360, ida y vuelta de ejercicios y feedback. Puede llegar a aportar mucho al sector.”

10.4. Cumplimiento de atributos de calidad

En esta sección se detallará el cumplimiento de los RNF correspondientes a atributos de calidad, definidos en la sección [6.3. Requerimientos no funcionales](#). Es importante destacar que el cumplimiento de los RNF 1, 2, 3 y 4, es justificado en la sección [7.2. Atributos de calidad](#).

- **RNF5 – Cumplimiento de las heurísticas de Nielsen**

A medida que se fueron generando las UI, se tuvieron en cuenta las heurísticas y se hizo un listado de evidencia de cumplimiento de las mismas. Para ver en detalle esto, ir a [13.22. Anexo 22: Cumplimiento de las Heurísticas de Nielsen \(RNF5\)](#).

- **RNF6 – La plataforma debe ser responsive a fin de garantizar la adecuada visualización en múltiples dispositivos**

Para el cumplimiento de esto, se utilizaron componentes provistos por la librería material-ui tales como Grid, Box, TextField, Button y funcionalidades como los breakpoints, que nos permitieron garantizar una interfaz responsive para los tamaños establecidos en el criterio de aceptación de este RNF. Para ver la evidencia de este cumplimiento ver [13.23. Anexo 23: Cumplimiento de responsiveness \(RNF6\)](#).

- **RNF7 – La plataforma debe seguir un sistema de diseño (Material Design)**

Como se detalló en la sección [7.3.1. Sistema de diseño](#), nuestra plataforma cumple con el sistema de diseño Material Design.

- **RNF8 – La plataforma deberá ser compatible con los navegadores Google Chrome y Safari, en sus versiones de escritorio y móviles (iOS y Android)**

A lo largo de toda la etapa de construcción, utilizamos estos navegadores para ir desarrollando nuestra plataforma. Por esto, podemos garantizar el cumplimiento de este RNF. Para ver evidencia ir a [13.24. Anexo 24: Cumplimiento de compatibilidad con navegadores \(RNF8\)](#).

10.5. Bugs

Luego de cada Release, procedíamos a verificar el cumplimiento de los criterios de aceptación correspondientes a las User Stories realizadas en dicho Release, a la vez que también testeábamos que las anteriores User Stories continuarán cumpliéndose. Como resultado de esto, comenzamos a identificar bugs como tareas en el Backlog. Para esto, agregamos una descripción del mismo, junto con su origen, prioridad, estimación de Story Points y quien lo reportó, como se ve en la Figura 53.

The screenshot shows a Jira issue page for '[FE] Evitar renderizado de los reproductores de video'. The issue is assigned to Santiago Tonarelli, has a story point estimate of 5, and a medium priority. It is linked to TBMJIRA-42, which is marked as 'DONE'. The description includes a bullet point: 'Evitar el renderizado de los reproductores cuando se cambie de tab, se active el quiz o se muestre la guía.' and another bullet point: 'US de la cual surge:'. The activity section shows 'Comments' and 'History' tabs.

Figura 53 -Tarea correspondiente a un bug detectado

Es importante destacar en nuestras Sprint Plannings realizábamos un análisis con el fin de seleccionar los bugs a tratar en el Sprint siguiente, teniendo en cuenta su estimación en SP, su prioridad (Alta, Media o Baja) y el contexto en el que nos encontrábamos.

A continuación, mostramos una tabla conteniendo los bugs detectados a lo largo del desarrollo, indicando su identificador, su nombre, su prioridad (P), el Release de origen, el Sprint en que se comenzó a trabajar en el bug y el Sprint en el cual se terminó (Tabla 24).

ID	Nombre	P	Origen	Inicio	Fin
TBMJIRA-46	[BE] Dependencia cíclica.	Alta	Release 1	Sprint 09	Sprint 09
TBMJIRA-47	[FE] Video de un curso carga un video de otro curso	Alta	Release 1	Sprint 09	Sprint 09
TBMJIRA-48	[FE] En prod se visualiza el código fuente del proyecto [PROD]	Media	Release 1	Sprint 09	Sprint 09

ID	Nombre	P	Origen	Inicio	Fin
TBMJIRA-54	[BE] No funciona el debug de tests unitarios	Media	Release 1	Sprint 10	Sprint 10
TBMJIRA-55	[BE] Links dentro de un curso no pueden repetirse	Alta	Release 1	Sprint 10	Sprint 10
TBMJIRA-56	[FE][BE] Descarga de archivos pdf no funciona	Media	Release 1	Sprint 10	Sprint 11
TBMJIRA-57	[FE] Reproductores de videos se ven desalineados cuando se muestra más de un ángulo de cámara	Media	Release 1	Sprint 10	Sprint 11
TBMJIRA-60	[BE] MediaConvert: se crean dos jobs al convertir un video	Media	Release 1	Sprint 11	Sprint 11
TBMJIRA-61	[BE] No se permite crear un curso con más de un link	Alta	Release 1	Sprint 11	Sprint 11
TBMJIRA-82	[BE] Validar que no se repita el "time" dentro de un mismo quiz	Media	Release 2	Sprint 12	Sprint 12
TBMJIRA-83	[FE] Duración de las lecciones y duración total se muestran en segundos	Baja	Release 2	Sprint 12	Sprint 12
TBMJIRA-84	[FE] Preguntas de recorrido virtual no se ven en pantalla completa	Baja	Release 2	Sprint 12	Sprint 12
TBMJIRA-85	[FE] No se puede acceder a los documentos (materiales) de un curso	Media	Release 2	Sprint 12	Sprint 12
TBMJIRA-96	[CD] Configurar CloudFront para distintos ambientes	Alta	Release 2	Sprint 13	Sprint 13
TBMJIRA-97	[FE][BE] No funciona el agregar opinión en vista previa de un curso	Media	Release 2	Sprint 13	Sprint 13
TBMJIRA-111	[FE] Bloquear DevTools en prod	Media	Release 3	Sprint 14	Sprint 14
TBMJIRA-112	[BE] Setear dinámicamente IDs de preguntas de un quiz de una lección	Baja	Release 3	Sprint 14	Sprint 14
TBMJIRA-121	[CD] Corregir permisos de S3 para subida de ejercicios	Media	Release 3	Sprint 15	Sprint 15
TBMJIRA-122	[FE] Evitar renderizado de los reproductores de video	Media	Release 3	Sprint 15	Sprint 15
TBMJIRA-140	[FE] Se tiene que recargar la página la primera vez que se ingresa a un vídeo o recorrido virtual de un curso para poder visualizarlo	Baja	Release 4	Sprint 16	Sprint 16
TBMJIRA-141	[FE] No se puede cambiar los ángulos de cámara desde el móvil	Baja	Release 4	Sprint 16	Sprint 16
TBMJIRA-142	[FE] Al recibir feedback de un ejercicio, si se ingresa desde los detalles del curso al ejercicio (estudiante), aparece como "pendiente de feedback"	Media	Release 4	Sprint 16	Sprint 16

ID	Nombre	P	Origen	Inicio	Fin
TBMJIRA-167	[FE] Mostrar los console log solo en DEV	Baja	Release 5	Sprint 17	Sprint 17
TBMJIRA-168	[FE] Recortar imagen del logo	Baja	Release 5	Sprint 17	Sprint 17

Tabla 24 - Bugs detectados durante el desarrollo

10.6. Conclusiones

En conclusión, la gestión de la calidad desempeñó un papel fundamental en el seguimiento y evaluación del nivel de calidad de nuestro producto. El uso de prácticas de aseguramiento de la calidad, junto con las validaciones realizadas y el cumplimiento de atributos de calidad, nos permiten afirmar que pudimos lograr un producto de calidad. Sin estas prácticas, no hubiera sido posible garantizar esto durante el transcurso de desarrollo del proyecto.

Asimismo, la gestión de la calidad también nos permitió fomentar una cultura de mejora continua en todo el equipo, lo que a su vez contribuyó a la entrega de un producto final de calidad y satisfacción de nuestros interesados.

11. Conclusiones finales

Para dar cierre a este documento, detallaremos las conclusiones generales a las cuales llegamos en base a lo realizado justificando el cumplimiento de nuestros objetivos. Luego, las lecciones aprendidas durante el proceso y cuáles serán nuestros próximos pasos.

11.1. Conclusiones generales

Desde un inicio, este proyecto fue un desafío para ambos integrantes, ya que se trataba de un proyecto de fin de carrera y a su vez de tipo emprendedor. Por otro lado, la duración del mismo a lo largo de más de un año, contando a su vez la etapa de postulación previa al inicio del proyecto fue algo nuevo para ambos integrantes. Ya que resultó sumamente desgastante, sumado a las tareas diarias de nuestros trabajos como desarrolladores y docentes.

El hecho de no poseer un cliente real y tener que depender de un público objetivo y de la disponibilidad de los interesados presento una experiencia única para ambos integrantes, teniendo que insistir en la comunicación y coordinación de reuniones, aun sabiendo que estos no contaban con la obligación de realizarlas.

Al tener un correcto manejo de los riesgos nos permitió gestionar el desarrollo de nuestro proyecto, evitando así tratar de forma correcta estos.

La arquitectura fue un desafío constante para ambos integrantes, ya que se buscó que esta cumpliera de la mejor manera posible con los objetivos, RF y RNF de nuestro proyecto.

Es importante destacar que nos encontramos muy conformes con el resultado obtenido en base al esfuerzo dedicado a lo largo de todo nuestro proyecto.

También, destacar que para nosotros fue clave la enseñanza y aprendizajes provisto por los expertos del área, ya que ninguno de nosotros tenía conocimientos acerca del sector.

Respecto a los objetivos definidos, a continuación detallaremos como cumplimos con los objetivos definidos.

- **Aplicar prácticas de Ingeniería de Software aprendidas durante la carrera**

Este objetivo se cumplió ya que a lo largo del transcurso del proyecto se usaron metodologías ágiles como Lean Startup y SCRUM y se hizo una ingeniería de requerimientos en base a un relevamiento de datos previo. A su vez, se creó una arquitectura de nuestra plataforma web siguiendo estándares, buenas prácticas y atributos de calidad vistos a lo largo de la carrera. También se realizaron una gestión del proyecto, de la calidad y de la configuración, utilizando distintas prácticas vistas en la carrera como se detalló a lo largo del documento.

- **Poner en práctica una metodología ágil vista en la carrera**

En la materia Ingeniería de Software Ágil, vimos en profundidad varios métodos ágiles, entre ellas SCRUM, la cual aplicamos con modificaciones debido a las características de nuestro proyecto.

- **Validar que la idea aporte valor a nuestro público objetivo a través de la creación de un MVP**

Este objetivo fue cumplido a través de la realización de validaciones con resultados satisfactorios con los interesados, lo cual se describe a lo largo del documento. A su vez, al finalizar el proceso de validaciones y teniendo todos los RF desarrollados, el comentario por parte de todos los interesados fue la necesidad de incluir los RF extra como parte del MVP ya que para ellos el valor que aportan estas funcionalidades ayudan a la cohesión de la plataforma junto con el resto de los RF ya implementados.

Entonces, por lo mencionado anteriormente, podemos decir que como resultado de la validación final, debemos incluir a la definición inicial de nuestro MVP los RF extra, para cumplir de manera eficiente con el objetivo del MVP; obtener información valiosa de usuarios reales que nos permitan validar la necesidad en el mercado.

- **Construcción de una plataforma web de calidad**

Con el uso de estándares, atributos de calidad y buenas prácticas mencionados en los diversos capítulos del documento, podemos afirmar que este objetivo fue cumplido.

11.2. Lecciones aprendidas

Luego de realizar el proyecto, pudimos comprender la complejidad de realizar un proyecto emprendedor y que la organización y documentación es un factor clave para llevarlo adelante. El rol del alcance es sumamente importante en este tipo de proyecto, ya que limitó el manejo y administración del emprendimiento. Gracias a que esta limitante fue controlada a tiempo, no afectó el resultado del mismo.

La experiencia en el manejo de metodologías y gestión de proyecto, entre otras actividades llevadas a cabo, es un factor clave que permite a los integrantes optimizar el tiempo de dedicación, así como su distribución eficiente para llevar adelante el proyecto. Creemos que, si volviéramos a tener la oportunidad de realizar un proyecto de estas características, podríamos hacerlo en un tiempo menor.

11.3. Próximos pasos

Teniendo en cuenta los resultados obtenidos, listaremos una serie de próximos pasos a realizar para continuar con el desarrollo y la mejora del emprendimiento.

- Retomar las comunicaciones con empresas constructoras y posibles interesados que nos ayuden en la validación de nuestra plataforma web con usuarios para poder recolectar aún más datos y en la formación de convenios.
- Retomar el contacto con el CIE, de modo de tener reuniones frecuentes con ellos con la finalidad de mantener su mentoría y apoyo en nuestro emprendimiento.
- Continuar con la postulación a fondos de inversión de la ANII, a modo de conseguir capital para financiar actividades de marketing y creación de cursos con profesionales del rubro.
- Desarrollar un mínimo producto comercializable (MMP) que nos permita llevar nuestra plataforma al siguiente nivel.
- Utilizar algún mecanismo de desarrollo que permite embeber nuestra plataforma web en una aplicación nativa para dispositivos móviles iOS y Android.

- Validar el modelo de ingresos de nuestra plataforma, de modo de garantizar la generación de ingresos.
- Analizar en más profundidad la aplicación de nuestra plataforma web en otros países de Latinoamérica.
- Hacer un estudio de costos de los servicios de AWS para nuestra plataforma web en profundidad.

12. Referencias bibliográficas

- [1] El Observador, “El empleo de la construcción aumentó 17% el último año y superó los 50.000 cotizantes”, 2022 [Online]. Disponible: <https://www.elobservador.com.uy/nota/el-empleo-en-la-industria-de-la-construccion-aumento-17-el-ultimo-ano--2022691850>
- [2] L. Bass, P. Clements, R. Kazman, *Software Architecture in Practice*, 3rd ed., Boston: Addison-Wesley, 2015.
- [3] CEEIC, “Relevancia del sector de la construcción en Uruguay”, 2019 [Online]. Disponible: <http://ceeic.uy/noticias/11-general/94-relevancia-del-sector-de-la-construccion-en-uruguay>
- [4] Smart Talent, “Servicios de arquitectura e ingeniería de Uruguay”, 2014 [Online]. Disponible: https://www.smarttalent.uy/innovaportal/file/848/1/informe_-_servicios_arq_e_ing_-_uruguay_xxi.pdf
- [5] Facultad de Ingeniería, “Datos de ingresos-egresos por carrera”, 2018 [Online]. Disponible: <https://www.fing.edu.uy/es/memorias-vivas/capitulo2b>
- [6] Massa Cinzenta, “Brasil tem 369 mil engenheiros civis e 202 mil arquitetos”, 2021 [Online]. Disponible: <https://www.cimentoitambe.com.br/massa-cinzenta/brasil-tem-369-mil-engenheiros-civis-e-202-mil-arquitetos/>
- [7] Global Market Insights, “Industry Trends”, 2022 [Online]. Disponible: <https://www.gminsights.com/industry-analysis/elearning-market-size>
- [8] Allied Market Research, “E-Learning Market Statistics: 2030”, 2021 [Online]. Disponible: <https://www.alliedmarketresearch.com/e-learning-market-A06253>
- [9] AWS, “Apéndice A: Diseño para la disponibilidad de determinados servicios de AWS”, [Online]. Disponible: https://docs.aws.amazon.com/es_es/wellarchitected/latest/reliability-pillar/appendix-a-designed-for-availability-for-select-aws-services.html
- [10] Mozilla Developer, “What is CSS?”, [Online]. Disponible: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS

[11] Andrés Mauro, “Impacto de los sistemas de diseño en la ingeniería de la interfaz de usuario del software”, Tesis de grado, Universidad ORT Uruguay, Montevideo, 2021 [Online]. Disponible:

<https://dspace.ort.edu.uy/bitstream/handle/20.500.11968/4474/Material%20completo.pdf?sequence=1&isAllowed=y>

[12] Material Design, “Cross-platform adaptation”, [Online]. Disponible:

<https://material.io/design/platform-guidance/cross-platform-adaptation.html#when-toadapt>

[13] Material-UI versión 4, [Online]. Disponible: <https://v4.mui.com/>

[14] Martina Severo y Santiago Tonarelli, “Tecnologías de software para interfaz de usuario adaptativa y multiplataforma”, Tesis de grado, Universidad ORT Uruguay, Montevideo, 2021 [Online]. Disponible:

<https://dspace.ort.edu.uy/bitstream/handle/20.500.11968/4497/Material%20completo.pdf?sequence=1&isAllowed=y>

[15] Material Design, “Color Tool”, [Online]. Available:

<https://material.io/resources/color/#!/?view.left=0&view.right=0>

[16] Redux, [Online]. Disponible: <https://redux.js.org/>

[17] GitHub, “flux”, [Online]. Disponible: <https://github.com/facebookarchive/flux>

[18] R. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, University of California, Irvine, 2000. [Online].

Available: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

[19] M. Masse, *REST API Design Rulebook*, CA: O'Reilly, 2011.

[20] 12factor, “The Twelve-Factor App”, 2017 [Online]. Disponible:

<https://12factor.net/>

[21] AWS, “Applying the Twelve-Factor App Methodology to Serverless Applications”, 2018 [Online]. Disponible: [https://aws.amazon.com/blogs/compute/applying-the-](https://aws.amazon.com/blogs/compute/applying-the-twelve-factor-app-methodology-to-serverless-applications/)

[twelve-factor-app-methodology-to-serverless-applications/](https://aws.amazon.com/blogs/compute/applying-the-twelve-factor-app-methodology-to-serverless-applications/)

- [22] AWS, “Quotas”, [Online]. Disponible: <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/cloudfront-limits.html>
- [23] Microsoft, “Multi-tenant SaaS database tenancy patterns”, 2023 [Online]. Disponible: <https://learn.microsoft.com/en-us/azure/azure-sql/database/saas-tenancy-app-design-patterns?view=azuresql>
- [24] TypeScript, “TypeScript Documentation”, [Online]. Disponible: <https://www.typescriptlang.org/docs/>
- [25] AWS, “AWS Documentation”, [Online]. Disponible: <https://docs.aws.amazon.com/index.html>
- [26] P. Clements, F. Bachmann, L. Bass, D. Garlan, and J. Ivers, *Documenting Software Architectures: Views and Beyond*, 2nd ed., Boston: Addison-Wesley, 2010.
- [27] UML, Página oficial, [Online]. Disponible: <https://www.uml.org/>
- [28] MultiThreaded, “Patterns of SOA: Asynchronous Transaction”, 2017 [Online]. Disponible: <https://multithreaded.stitchfix.com/blog/2017/05/09/patterns-of-soa-asynchronous-transaction/>
- [29] AWS, “AWS Activate Founders”, [Online]. Disponible: <https://aws.amazon.com/es/activate/founders/>
- [30] Atlassian, “Flujo de trabajo de Gitflow”, [Online]. Disponible: <https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow>

13. Anexos

13.1. Anexo 1: Entrevistas realizadas

A continuación, detallaremos las preguntas realizadas de las entrevistas a múltiples trabajadores del sector, como mencionamos en la sección [4.1.1. Entrevistas](#). A modo de guiarnos, creamos una plantilla con una serie de preguntas a seguir al momento de llevar a cabo cada entrevista. Igualmente, dependiendo de cada entrevistado, las preguntas pueden haber variado en el momento para adaptarse a las respuestas que íbamos recibiendo. Las preguntas fueron las siguientes:

- 1. *¿Qué conocimientos que tuviste en facultad te gustaría reforzar?***
- 2. *¿Qué conocimientos no son impartidos o adquiridos en instituciones educativas?***
- 3. *¿Qué tanta importancia tiene un diploma o certificación como respaldo?***
- 4. *¿Qué nivel de educación se debería tener para poder realizar algún tipo de curso específico/trabajo?***
- 5. *¿Existe alguna certificación directa solo haciendo alguna prueba, sin necesidad de hacer un curso?***
- 6. *¿Qué tipo de materiales utilizas para reforzar conocimientos previos?***
- 7. *¿Qué herramientas usas para actualizar tu conocimiento acorde a las nuevas prácticas, metodologías y herramientas?***
- 8. *¿Has tenido experiencias con clases/cursos virtuales?***
- 9. *¿Qué desventajas le encuentras a las clases/cursos virtuales frente a los presenciales?***
- 10. *De los temas que manejas, ¿te sentís confiado para explicar/exponer alguno?***
- 11. *¿Qué tanto utilizas la tecnología en el trabajo?***
- 12. *¿Con cuánto tiempo libre contás fuera del ámbito laboral?***

Se optó por pasar las notas tomadas a partir de las respuestas a una referencia puesto que ampliaría considerablemente la extensión del documento. La información puede encontrarse en la referencia [1].

Referencias bibliográficas

[1] Google Docs, “Entrevistas”, 2022 [Online]. Disponible: https://docs.google.com/document/d/1rGqdOwbFgRdNjL_laitoI0B6NAiSpdB26ZGLW_NbPR4E/edit?usp=sharing.

13.2. Anexo 2: Encuesta realizada

A continuación, mostramos la encuesta llevada a cabo a diversos trabajadores del sector de la construcción, contando con un total de 128 respuestas, como mencionamos en la sección [4.1.2. Encuestas](#). Es importante destacar que algunas de las preguntas eran opcionales, por lo que la cantidad de respuestas por pregunta puede variar.

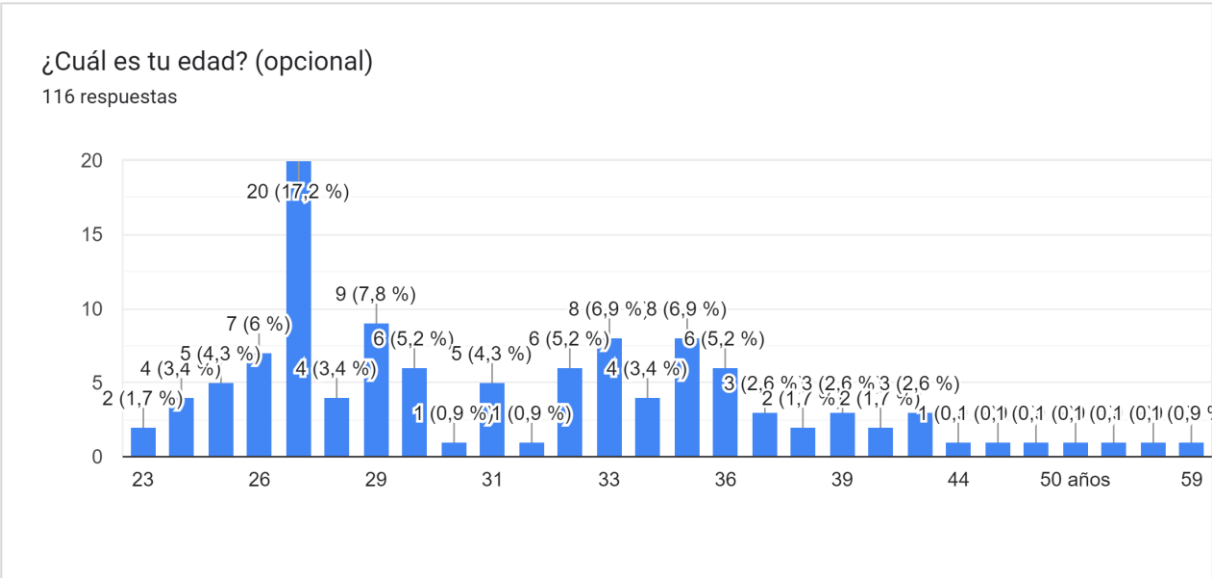


Figura 54 - Edad de los encuestados

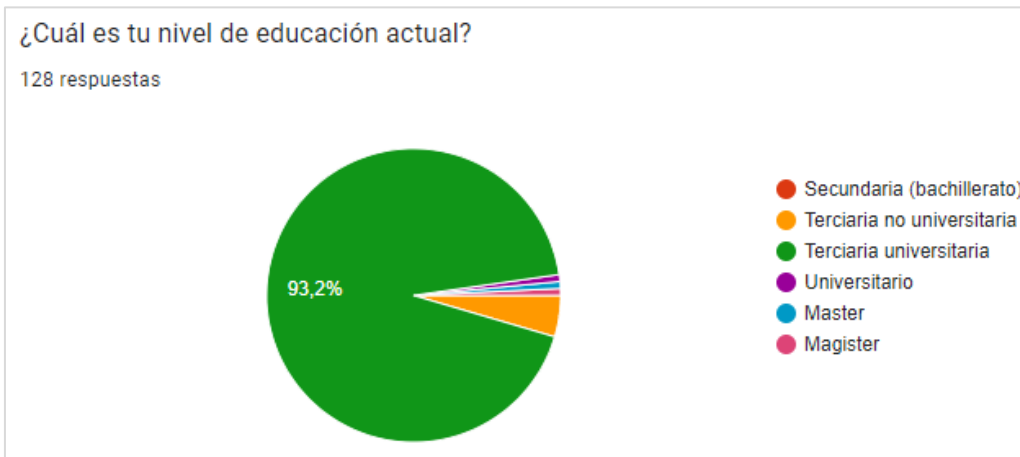


Figura 55 - Nivel de educación de los encuestados

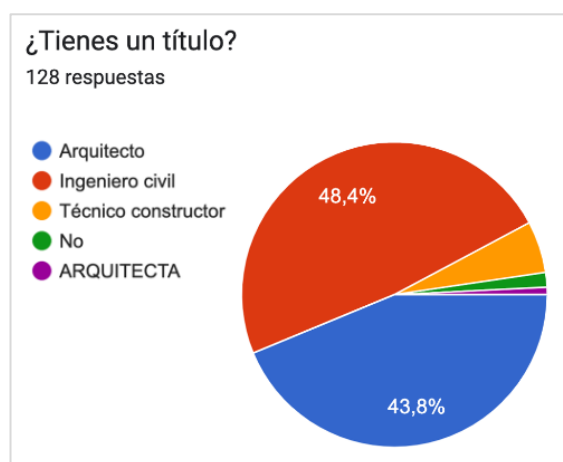


Figura 56 - Título de los encuestados



Figura 57 - Capacitación de los encuestados

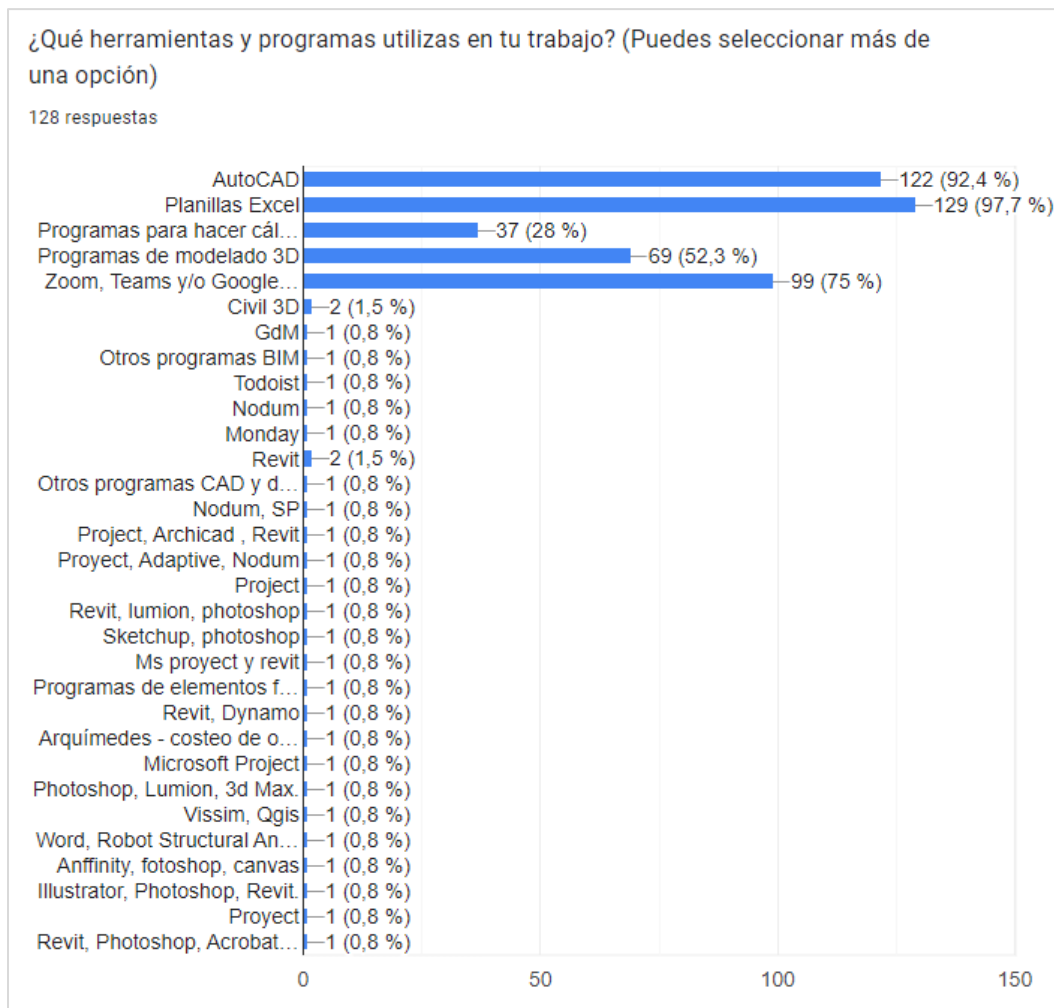


Figura 58 - Herramientas más utilizados por los encuestados



Figura 59 - Experiencia con cursos virtuales de los encuestados



Figura 60 - Necesidad de distintos puntos de vista de los encuestados

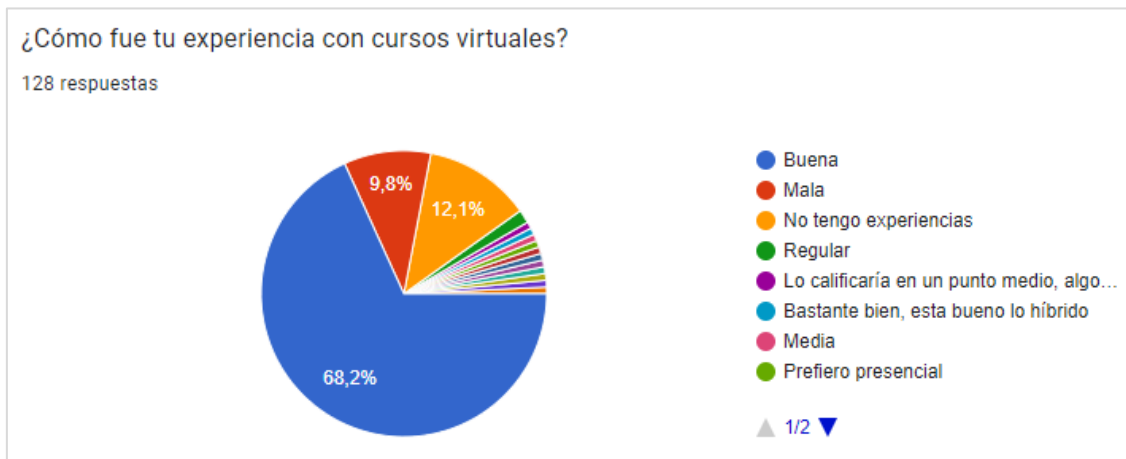


Figura 61 - Experiencia personal con cursos virtuales



Figura 62 - Enfoque de los cursos

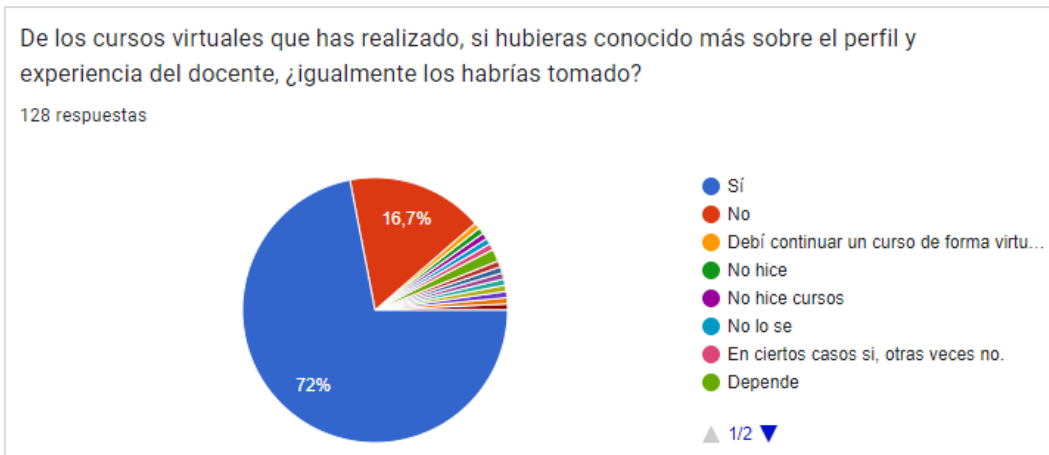


Figura 63 - Impacto del conocimiento del docente del curso



Figura 64 - Reseñas de los cursos

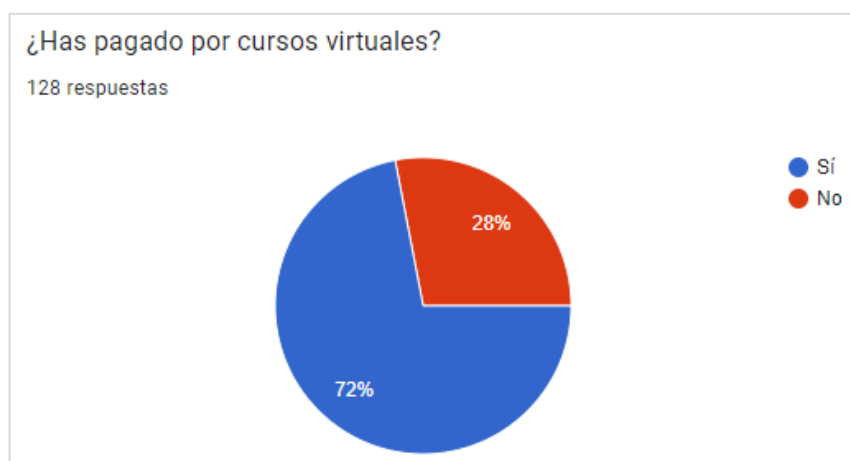


Figura 65 - Compra de cursos virtuales

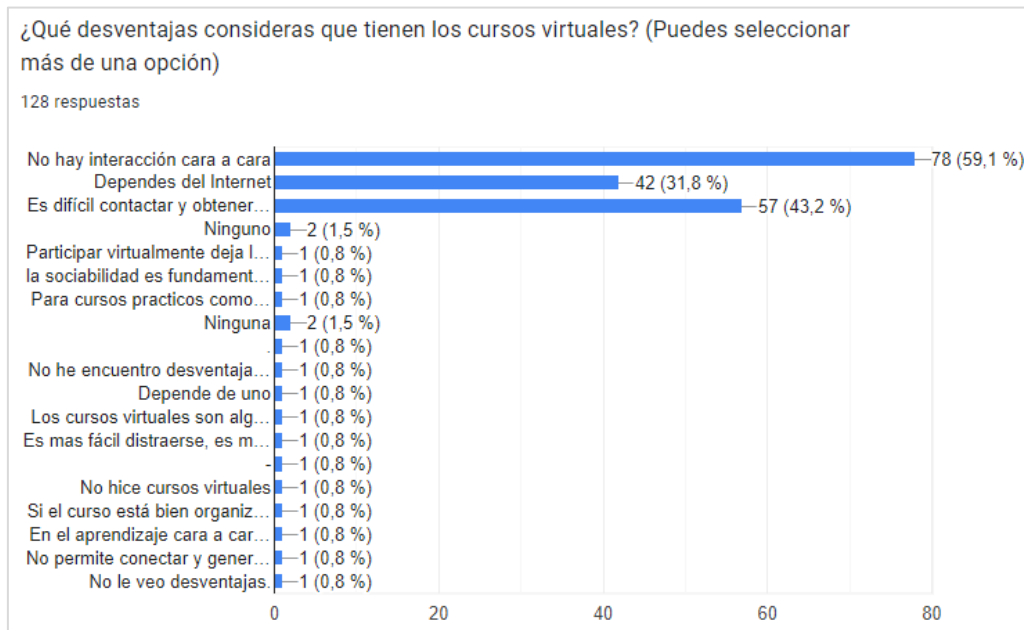


Figura 66 - Desventajas de los cursos virtuales

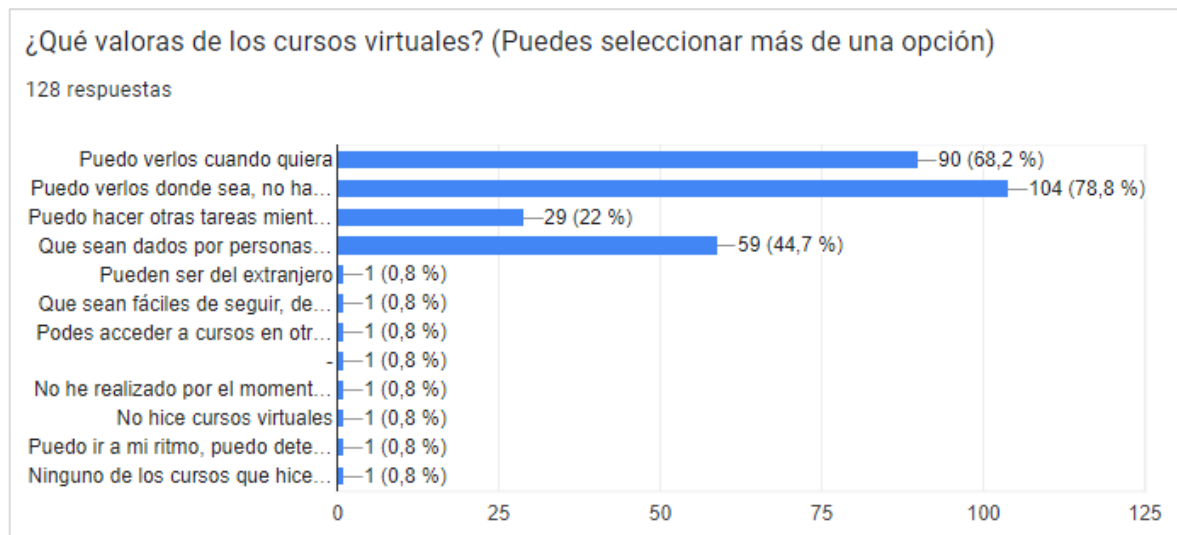


Figura 67 - Valoración de los cursos virtuales

De los cursos virtuales que has realizado, ¿qué tanto conocías de la persona que lo dictaba?

128 respuestas



Figura 68 - Conocimiento del docente

¿Estás interesado/a en tomar cursos virtuales pagos, que sean grabados por personas especializadas y con experiencia en el rubro de la construcción, sobre temas específicos?

128 respuestas

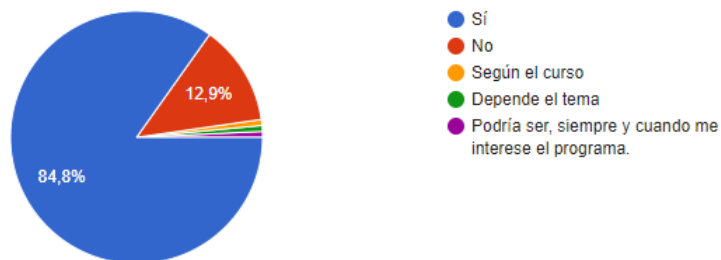


Figura 69 - Interés en tomar cursos virtuales

¿Estás dispuesto/a a brindar cursos, grabándote explicando temas, herramientas y/o procesos particulares (de tu elección) relacionados al rubro de la construcción, obteniendo dinero por ello?

128 respuestas



Figura 70 - Disposición a brindar cursos virtuales

¿Sobre qué temas podrías enseñar?

128 respuestas

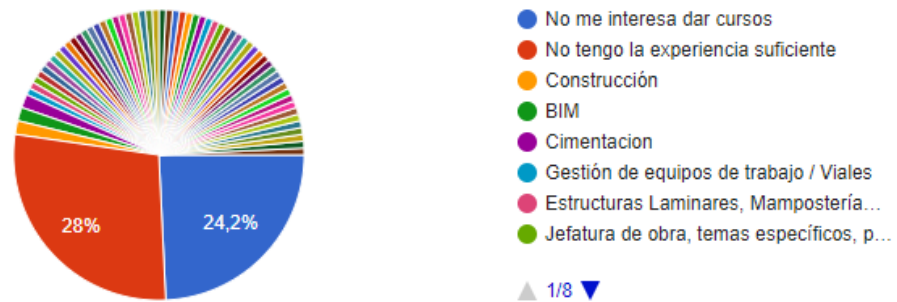


Figura 71 - Temas para enseñar

13.3. Anexo 3: Documentación CIE

Se escogió pasar esta documentación a una referencia puesto que es otro documento con distinto formato al presente y por temas relativos a la extensión de este documento. La información puede encontrarse en la referencia [1].

Referencias bibliográficas

[1] Google Drive, "Formulario adicional para Proyectos emprendedores", 2022

[Online]. Disponible:

https://drive.google.com/file/d/1K2avs8FlqjEiHsiPSTWqIMtB_Qr6Y3Ys/view?usp=sharing

13.4. Anexo 4: Prototipos realizados

Se optó por pasar estos prototipos, que creamos a partir de los insights obtenidos del relevamiento de datos, a través del uso de la herramienta Whimsical, a una referencia debido a su extensión. La información puede encontrarse en la referencia [1].

Referencias bibliográficas

[1] Whimsical, "TBM Prototypes", 2022 [Online]. Disponible:

<https://whimsical.com/tbm-prototypes-RVWaiYwvASkjaYofxjCvD>

13.5. Anexo 5: RF iniciales

A continuación, se muestran los RF iniciales, mencionados en la sección [6.1.2. Priorización y reescritura](#), escritos en formato de narrativa (Figura 72).

<p>RF1 - Como usuario Quiero poder crear cursos, incluyendo subida de videos, archivos y ejercicios entregables Para poder compartir mis conocimientos.</p>	<p>RF2 - Como usuario, Quiero poder ver videos con opción de distintos ángulos de cámara Para aprender de forma más eficiente.</p>	<p>RF3 - Como usuario Quiero poder buscar cursos, ver un catálogo de cursos separados por temas de una obra, ver recomendaciones de cursos en base a mis gustos y preferencias Para que mi experiencia sea más personalizada y adecuada a mi persona.</p>
<p>RF4 - Como usuario Quiero poder seleccionar cursos virtuales como favoritos Para poder acceder a ellos más rápidamente.</p>	<p>RF5 - Como usuario Quiero poder hacer recorridos virtuales de obras Para tener experiencias realistas sin tener que desplazarme físicamente hasta el lugar.</p>	<p>RF6 - Como usuario Quiero tener un perfil, incluyendo nombre, apellido, formación, situación laboral, foto y descripción Para autenticar mi identidad y brindar más información acerca de mí.</p>
<p>RF7 - Como usuario Quiero poder subir ejercicios entregables Para poder practicar lo aprendido y recibir feedback.</p>	<p>RF8 - Como usuario Quiero poder brindar feedback de ejercicios Para que los alumnos reciban comentarios acerca de su progreso.</p>	<p>RF9 - Como usuario Quiero poder corregir ejercicios Para conseguir recompensas.</p>
<p>RF10 - Como usuario Quiero poder tener un canal de comunicación directa con los docentes Para consultar dudas.</p>	<p>RF11 - Como usuario Quiero poder comprar cursos virtuales Para poder verlos de manera indefinida y cuando lo desee.</p>	<p>RF12 - Como usuario Quiero tener una sección de ayuda Para poder resolver cualquier inconveniente que puedan llegar a tener con la plataforma.</p>
<p>RF13 - Como usuario Quiero tener un canal con ofertas laborales de empresas constructoras Para estar al tanto de los empleos disponibles.</p>		

Figura 72 - RF iniciales en formato de narrativa

13.6. Anexo 6: Desglose de US

A continuación, detallamos todas las User Stories (US) creadas con el formato mencionado en la sección [6.1.3. Especificación](#).

RF1 – Cursos	
Título: TBMJIRA-8 Videos de un curso	Estimación: 51 SP
<p>User Story:</p> <p>Como usuario Quiero poder ver los videos de un curso Para profundizar conocimientos sobre un tema en específico.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-8-AC-1: Visualización de video de un curso exitosa</u> Dado un usuario que quiere ver un video de un curso Cuando hace clic en el respectivo video Y hace clic en el botón de reproducir Entonces el video comienza a reproducirse.</p> <p><u>TBMJIRA-8-AC-2: Visualización de video de un curso fallida sin conexión a Internet</u> Dado un usuario que quiere ver un video de un curso Cuando hace clic en el respectivo video Y hace clic en el botón de reproducir Entonces el sistema muestra un mensaje informativo en pantalla Y el video no se reproduce.</p>	

Tabla 25 - Detalle de US TBMJIRA-8

Tareas contenidas en TBMJIRA-8 Videos de un curso		
ID	Nombre	SP
TBMJIRA-9	[FE] Instalación de librerías básicas	2
TBMJIRA-10	[BE] Instalación de librerías básicas	2
TBMJIRA-11	[CD] Configuración de AWS (IAM, usuarios y roles)	8
TBMJIRA-12	[CD] Configuración de serverless para deploy	5
TBMJIRA-13	[FE] Creación de barra de menú	5
TBMJIRA-14	[BE] Creación de endpoint para obtener información de un video de un curso	13
TBMJIRA-15	[CD] Creación del bucket en S3	3
TBMJIRA-16	[FE] Creación de la pantalla	8
TBMJIRA-17	[BE] Creación de test para el endpoint	5

Tabla 26 - Tareas dentro de TBMJIRA-8

RF1 – Cursos	
Título: TBMJIRA-18 División en módulos de un curso	Estimación: 49 SP
<p>User Story:</p> <p>Como usuario Quiero entrar a un curso y ver su contenido Para poder entender la división del mismo y cómo está organizado su material.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-18-AC-1: Visualización de módulos con videos, ejercicios y archivos exitosa</u> Dado un curso Cuando entro a ver su contenido Entonces veo los módulos con videos, archivos y ejercicios correspondientes al curso.</p> <p><u>TBMJIRA-18-AC-2: Visualización de módulos con videos, ejercicios y archivos fallida sin conexión a Internet</u> Dado un curso Cuando entro a ver su contenido Entonces el sistema informa que no se pudieron cargar los módulos de un curso.</p>	

Tabla 27 - Detalle de US TBMJIRA-18

Tareas contenidas en TBMJIRA-18 División en módulos de un curso		
ID	Nombre	SP
TBMJIRA-19	[BE] Configuración de AWS CLI	5
TBMJIRA-20	[BE] Construcción de arquitectura a gran escala	8
TBMJIRA-21	[FE] Construcción de arquitectura a gran escala	8
TBMJIRA-22	[BE] Configuración de CI y CD	5
TBMJIRA-23	[FE] Configuración de CI y CD	5
TBMJIRA-24	[CD] Costos de CI y CD en AWS para BE provocaron que se tuvieran que hacer con GitHub Actions	5
TBMJIRA-25	[BE] Creación de endpoint para obtener información de un curso	5
TBMJIRA-26	[FE] Creación de la pantalla	5
TBMJIRA-27	[BE] Creación de test para el endpoint	3

Tabla 28 - Tareas dentro de TBMJIRA-18

RF1 – Cursos	
Título: TBMJIRA-28 Distintos ángulos de cámara de un curso	Estimación: 8 SP
<p>User Story:</p> <p>Como usuario Quiero poder ver los videos de un curso pudiendo elegir los ángulos de cámara a mostrar Para tener un aprendizaje rápido y útil mediante varios enfoques.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-28-AC-1: Visualización de varios ángulos de cámara de un video de un curso exitosa</u> Dado un usuario que quiere ver más de un ángulo de un video de un curso Cuando hace clic en la cantidad de ángulos que desea que se muestren Y hace clic en el botón de aplicar Entonces se muestran los ángulos de cámara correspondientes.</p> <p><u>TBMJIRA-28-AC-2: Visualización de varios ángulos de cámara de un video de un curso fallida sin conexión a Internet</u> Dado un usuario que quiere ver más de un ángulo de un video de un curso Cuando hace clic en la cantidad de ángulos que desea que se muestren Y hace clic en el botón de aplicar Entonces el sistema muestra un mensaje informativo en pantalla Y no se muestran los ángulos de cámara seleccionados.</p>	

Tabla 29 - Detalle de US TBMJIRA-28

Tareas contenidas en TBMJIRA-28 Distintos ángulos de cámara de un curso		
ID	Nombre	SP
TBMJIRA-29	[FE] Agregación de varios reproductores de video	8

Tabla 30 - Tareas dentro de TBMJIRA-28

RF1 – Cursos	
Título: TBMJIRA-30 Vista previa de un curso	Estimación: 16 SP
<p>User Story:</p> <p>Como usuario Quiero ver los detalles relacionados a un curso Para saber más acerca del mismo y decidir si el curso es de mi interés.</p>	

Criterios de aceptación (escenarios):

TBMJIRA-30-AC-1: Visualización previa de un curso exitosa

Dada la lista de cursos existentes

Cuando se ingresa a ver los detalles de uno

Entonces se ve su información, junto con una descripción, lista de contenido y opiniones de otros usuarios.

TBMJIRA-30-AC-2: Visualización previa de un curso fallida sin conexión a Internet

Dada la lista de cursos existentes

Cuando se ingresa a ver los detalles de uno

Entonces el sistema informa que no se pudo realizar la operación.

Tabla 31 - Detalle de US TBMJIRA-30

Tareas contenidas en TBMJIRA-30 Vista previa de un curso		
ID	Nombre	SP
TBMJIRA-31	[FE] Creación de la pantalla	8
TBMJIRA-32	[BE] Creación de endpoint para obtener datos	5
TBMJIRA-33	[BE] Creación de test para el endpoint	3

Tabla 32 - Tareas dentro de TBMJIRA-30

RF1 – Cursos	
Título: TBMJIRA-34 Sección de preguntas de un curso	Estimación: 19 SP
User Story: Como usuario Quiero entrar a un curso y ver las preguntas de otros usuarios Para obtener información útil que otros usuarios hayan compartido y plantear interrogantes que tenga.	

Criterios de aceptación (escenarios):

TBMJIRA-34-AC-1: Visualización de preguntas de un curso exitosa

Dado un curso

Cuando se entra a ver las preguntas realizadas por otros usuarios

Entonces se ven las preguntas junto con información del usuario que las creó, con sus respectivas respuestas.

TBMJIRA-34-AC-2: Visualización de preguntas de un curso fallida sin conexión a Internet

Dado un curso

Cuando se entra a ver las preguntas realizadas por otros usuarios

Entonces el sistema informa que no se pudieron cargar las preguntas de un curso.

TBMJIRA-34-AC-3: Agregación de pregunta a un curso exitosa

Dado un curso

Cuando se entra a ver las preguntas realizadas por otros usuarios

Y se hace clic en el botón para agregar una pregunta

Entonces se muestran los campos de texto para ingresar título y cuerpo de la pregunta

Y el botón para crearla.

TBMJIRA-34-AC-4: Agregación de pregunta a un curso fallida sin conexión a Internet

Dado un curso

Cuando se entra a ver las preguntas realizadas por otros usuarios

Y se hace clic en el botón para agregar una pregunta

Entonces el sistema informa que no se pudo realizar la operación.

Tabla 33 - Detalle de US TBMJIRA-34

Tareas contenidas en TBMJIRA-34 Sección de preguntas de un curso		
ID	Nombre	SP
TBMJIRA-35	[FE] Creación de la pantalla	8
TBMJIRA-36	[BE] Creación de endpoint para obtener datos	8
TBMJIRA-37	[BE] Creación de test para el endpoint	3

Tabla 34 - Tareas dentro de TBMJIRA-34

RF1 – Cursos	
Título: TBMJIRA-38 Archivos informativos de un curso	Estimación: 16 SP
User Story: Como usuario Quiero entrar a un curso y dentro de su contenido, ver sus archivos informativos Para poder visualizar material adjunto e información útil relacionada al mismo.	

Criterios de aceptación (escenarios):*TBMJIRA-38-AC-1: Visualización de archivos informativos de un curso exitosa***Dado** un curso**Cuando** se entra a ver el contenido un curso**Y** se hace clic en un archivo informativo**Entonces** se muestra el material correspondiente.*TBMJIRA-38-AC-2: Visualización de archivos informativos de un curso fallida sin conexión a Internet***Dado** un curso**Cuando** se entra a ver el contenido un curso**Y** se hace clic en un archivo informativo**Entonces** el sistema informa que no se pudo cargar el archivo.

Tabla 35 - Detalle de US TBMJIRA-38

Tareas contenidas en TBMJIRA-38 Archivos informativos de un curso		
ID	Nombre	SP
TBMJIRA-39	[FE] Creación de la pantalla	8
TBMJIRA-40	[BE] Creación de endpoint para obtener datos	5
TBMJIRA-41	[BE] Creación de test para el endpoint	3

Tabla 36 - Tareas dentro de TBMJIRA-38

RF2 – Recorridos virtuales	
Título: TBMJIRA-42 Implementación de recorrido virtual usando video 360	Estimación: 76 SP
User Story:	
Como usuario	
Quiero poder ver un recorrido virtual	
Para tener la experiencia de recorrer una obra virtualmente.	
Criterios de aceptación (escenarios):	
<u><i>TBMJIRA-42-AC-1: Visualización de varios ángulos de cámara de un video de un curso exitosa</i></u>	
Dado un usuario que quiere ver más de un ángulo de un video de un curso	
Cuando hace clic en la cantidad de ángulos que desea que se muestren	
Y hace clic en el botón de aplicar	
Entonces se muestran los ángulos de cámara correspondientes.	
<u><i>TBMJIRA-42-AC-2: Visualización de varios ángulos de cámara de un video de un curso fallida sin conexión a Internet</i></u>	
Dado un usuario que quiere ver más de un ángulo de un video de un curso	
Cuando hace clic en la cantidad de ángulos que desea que se muestren	
Y hace clic en el botón de aplicar	
Entonces el sistema muestra un mensaje informativo en pantalla	
Y no se muestran los ángulos de cámara seleccionados.	

Tabla 37 - Detalle de US TBMJIRA-42

Tareas contenidas en TBMJIRA-42 Implementación de recorrido virtual usando video 360		
ID	Nombre	SP
TBMJIRA-43	[FE] Creación de componente de reproducción de video 360	34
TBMJIRA-44	[CD] Configuración de MediaConvert para reproducción de videos 360	21
TBMJIRA-45	[BE] Integración con servicios de video 360	21

Tabla 38 - Tareas dentro de TBMJIRA-42

RF2 – Recorridos virtuales	
Título: TBMJIRA-49 Preguntas a lo largo del recorrido virtual	Estimación: 39 SP
<p>User Story:</p> <p>Como usuario Quiero poder tener preguntas a lo largo de un recorrido virtual Para poner en práctica lo aprendido y hacer la experiencia más interactiva.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-49-AC-1: Visualización de preguntas a lo largo de un recorrido virtual exitosa</u> Dado un usuario que quiere tener preguntas a lo largo de un recorrido virtual Cuando está visualizando el video Entonces se muestran preguntas de Múltiple Opción y/o Verdadero o Falso cada cierto tiempo.</p> <p><u>TBMJIRA-49-AC-2: Visualización de preguntas a lo largo de un recorrido virtual fallida sin conexión a Internet</u> Dado un usuario que quiere tener preguntas a lo largo de un recorrido virtual Cuando está visualizando el video Entonces el sistema muestra un mensaje informativo en pantalla Y no reproduce el video ni se muestran las preguntas.</p>	

Tabla 39 - Detalle de US TBMJIRA-49

Tareas contenidas en TBMJIRA-49 Preguntas a lo largo del recorrido virtual		
ID	Nombre	SP
TBMJIRA-50	[FE] Crear el componente del modal para preguntas	13
TBMJIRA-51	[FE] Conectar modal con video 360	13
TBMJIRA-52	[BE] Asociar lista de preguntas a una lección de tipo recorrido virtual	8
TBMJIRA-53	[FE] Conectar el frontend con el backend	5

Tabla 40 - Tareas dentro de TBMJIRA-49

RF2 – Recorridos virtuales	
Título: TBMJIRA-58 Sección de explicación de funcionamiento del recorrido virtual	Estimación: 3 SP
<p>User Story:</p> <p>Como usuario Quiero tener una guía acerca del funcionamiento del recorrido virtual Para saber cómo funciona.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-58-AC-1: Visualización de explicación de funcionamiento de un recorrido virtual exitosa</u> Dado un usuario que quiere saber cómo funciona un recorrido virtual Cuando ingresa a la visualización del video Entonces se muestra una guía en alguna parte.</p> <p><u>TBMJIRA-58-AC-2: Visualización de explicación de funcionamiento de un recorrido virtual fallida sin conexión a Internet</u> Dado un usuario que quiere saber cómo funciona un recorrido virtual Cuando ingresa a la visualización del video Entonces el sistema informa que no se pudo cargar el contenido.</p>	

Tabla 41 - Detalle de US TBMJIRA-58

Tareas contenidas en TBMJIRA-58 Sección de explicación de funcionamiento del recorrido virtual		
ID	Nombre	SP
TBMJIRA-59	[FE] Creación de componente de guía para recorridos virtuales	3

Tabla 42 - Tareas dentro de TBMJIRA-58

RF3 – Ejercicios entregables	
Título: TBMJIRA-69 Registro de un usuario	Estimación: 21 SP
<p>User Story:</p> <p>Como usuario no registrado Quiero registrarme en la plataforma Para poder ingresar a la misma.</p>	

Criterios de aceptación (escenarios):*TBMJIRA-69-AC-1: Registro de usuario exitoso***Dado** un usuario no registrado**Cuando** hace clic en el botón para registrarse**Entonces** se muestra el form para ingresar los datos**Y** una vez enviado se le manda un código de verificación**Y** una vez lo confirma es registrado en el sistema.*TBMJIRA-69-AC-2: Registro de usuario fallido sin conexión a Internet***Dado** un usuario no registrado**Cuando** hace clic en el botón para registrarse**Entonces** el sistema informa que no se pudo cargar el contenido.

Tabla 43 - Detalle de US TBMJIRA-69

Tareas contenidas en TBMJIRA-69 Registro de un usuario		
ID	Nombre	SP
TBMJIRA-70	[FE] Creación de la pantalla	8
TBMJIRA-71	[FE] Conexión con el backend	5
TBMJIRA-72	[BE] Creación del endpoint	8

Tabla 44 - Tareas dentro de TBMJIRA-69

RF3 – Ejercicios entregables	
Título: TBMJIRA-73 Entrega de ejercicios entregables	Estimación: 8 SP
User Story: Como usuario Quiero entregar un ejercicio Para poder recibir feedback y verificar mi aprendizaje.	
Criterios de aceptación (escenarios): <u><i>TBMJIRA-73-AC-1: Entrega de ejercicios entregables exitosa</i></u> Dado un usuario Cuando ingresa a un ejercicio de un curso Y sube un archivo Y le da clic a entregar Entonces el sistema le notifica de su entrega Y el estado del ejercicio pasa a ser “entregado (pendiente de feedback)”. <u><i>TBMJIRA-73-AC-2: Entrega de ejercicios entregables fallida sin conexión a Internet</i></u> Dado un usuario Cuando ingresa a un ejercicio de un curso Entonces el sistema informa que no se pudo cargar el contenido.	

Tabla 45 - Detalle de US TBMJIRA-73

Tareas contenidas en TBMJIRA-73 Entrega de ejercicios entregables		
ID	Nombre	SP
TBMJIRA-74	[FE] Creación de la pantalla	2
TBMJIRA-75	[FE] Conexión con el backend	1
TBMJIRA-76	[BE] Creación del endpoint	5

Tabla 46 - Tareas dentro de TBMJIRA-73

RF3 – Ejercicios entregables	
Título: TBMJIRA-77 Detalle de ejercicio entregable de un curso	Estimación: 21 SP
<p>User Story:</p> <p>Como usuario Quiero entrar a un curso y dentro de su contenido, ver sus ejercicios entregables Para poder visualizar los detalles de ejercicios relacionados al mismo.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-77-AC-1: Visualización de ejercicios entregables de un curso exitosa</u> Dado un curso Cuando se entra a ver el contenido un curso Y se hace clic en un ejercicio entregable Entonces se muestran los detalles del mismo.</p> <p><u>TBMJIRA-77-AC-2: Visualización de ejercicios entregables de un curso fallida sin conexión a Internet</u> Dado un curso Cuando se entra a ver el contenido un curso Y se hace clic en un ejercicio entregable Entonces el sistema informa que no se pudo cargar el contenido.</p>	

Tabla 47 - Detalle de US TBMJIRA-77

Tareas contenidas en TBMJIRA-77 Detalle de ejercicio entregable de un curso		
ID	Nombre	SP
TBMJIRA-78	[FE] Creación de la pantalla	8
TBMJIRA-79	[FE] Conexión con el backend	5
TBMJIRA-80	[BE] Creación del endpoint	3
TBMJIRA-81	[BE] Conexión con la BD	5

Tabla 48 - Tareas dentro de TBMJIRA-77

RF3 – Ejercicios entregables	
Título: TBMJIRA-86 Visualización de consigna de ejercicio en lección	Estimación: 5 SP
<p>User Story:</p> <p>Como usuario Quiero entrar a un ejercicio y dentro de su contenido, ver la consigna Para poder saber de qué se trata.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-86-AC-1: Visualización de consigna de ejercicio entregable exitosa</u> Dado un ejercicio Cuando se ingresa a ver su contenido Y se hace clic en ver consigna Entonces se muestra la letra del ejercicio.</p> <p><u>TBMJIRA-86-AC-2: Visualización de consigna de ejercicio entregable fallida sin conexión a Internet</u> Dado un ejercicio Cuando se ingresa a ver su contenido Entonces el sistema informa que no se pudo cargar el contenido.</p>	

Tabla 49 - Detalle de US TBMJIRA-86

Tareas contenidas en TBMJIRA-86 Visualización de consigna de ejercicio en lección		
ID	Nombre	SP
TBMJIRA-87	[FE] Creación de componente	2
TBMJIRA-88	[CD] Acceso al link mediante CloudFront	3

Tabla 50 - Tareas dentro de TBMJIRA-86

RF3 – Ejercicios entregables	
Título: TBMJIRA-89 Descarga de ejercicio entregado	Estimación: 5 SP
<p>User Story:</p> <p>Como usuario Quiero poder descargar la entrega realizada en un ejercicio de un curso Para poder verificar que se haya subido correctamente.</p>	

Criterios de aceptación (escenarios):*TBMJIRA-89-AC-1: Descarga de una entrega realizada exitosa***Dado** un ejercicio**Cuando** se ingresa a ver su contenido**Y** se hace clic en ver entrega**Entonces** se muestra el contenido subido.*TBMJIRA-89-AC-2: Descarga de una entrega realizada fallida sin conexión a Internet***Dado** un ejercicio**Cuando** se ingresa a ver su contenido**Entonces** el sistema informa que no se pudo cargar el contenido.

Tabla 51 - Detalle de US TBMJIRA-89

Tareas contenidas en TBMJIRA-89 Descarga de ejercicio entregado		
ID	Nombre	SP
TBMJIRA-90	[FE] Creación de componente	2
TBMJIRA-91	[CD] Descarga del archivo desde CloudFront	3

Tabla 52 - Tareas dentro de TBMJIRA-89

RF3 – Ejercicios entregables	
Título: TBMJIRA-92 Listado de ejercicios entregables	Estimación: 21 SP
User Story: Como usuario Quiero ver el listado de mis ejercicios entregados Para poder verificar su estado y si fueron corregidos.	
Criterios de aceptación (escenarios): <u><i>TBMJIRA-92-AC-1: Listado de ejercicios entregables exitosa</i></u> Dado un conjunto de ejercicios entregados por un usuario Cuando se hace clic en la opción de menú correspondiente Entonces se muestra la lista de ejercicios entregados por el usuario. <u><i>TBMJIRA-92-AC-2: Listado de ejercicios entregables fallida sin conexión a Internet</i></u> Dado un conjunto de ejercicios entregados por un usuario Cuando se hace clic en la opción de menú correspondiente Entonces el sistema informa que no se pudo cargar el contenido.	

Tabla 53 - Detalle de US TBMJIRA-92

Tareas contenidas en TBMJIRA-92 Listado de ejercicios entregables		
ID	Nombre	SP
TBMJIRA-93	[FE] Creación de la pantalla	8
TBMJIRA-94	[FE] Conexión con el backend	5
TBMJIRA-95	[BE] Creación del endpoint	8

Tabla 54 - Tareas dentro de TBMJIRA-92

RF4 – Feedback	
Título: TBMJIRA-102 Listado de ejercicios entregados sin revisar	Estimación: 19 SP
<p>User Story:</p> <p>Como usuario responsable de un curso Quiero ver los ejercicios entregados por otros usuarios a mi curso Para para poder darles feedback acerca de lo que realizaron.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-102-AC-1: Listado de ejercicios entregados sin revisar exitoso</u> Dado un usuario responsable de un curso Cuando selecciona la opción del menú correspondiente a ejercicios Entonces se muestra la lista de ejercicios entregados por otros usuarios a su curso pendientes de feedback.</p> <p><u>TBMJIRA-102-AC-2: Listado de ejercicios entregados sin revisar fallido sin conexión a Internet</u> Dado un usuario responsable de un curso Cuando selecciona la opción del menú correspondiente a ejercicios Entonces el sistema informa que no se pudo cargar el contenido.</p>	

Tabla 55 - Detalle de US TBMJIRA-102

Tareas contenidas en TBMJIRA-102 Listado de ejercicios entregados sin revisar		
ID	Nombre	SP
TBMJIRA-103	[FE] Creación de componente	8
TBMJIRA-104	[FE] Conexión con el backend	3
TBMJIRA-105	[BE] Creación del endpoint	5
TBMJIRA-106	[CD] Agregar índice por identificador de docente	3

Tabla 56 - Tareas dentro de TBMJIRA-102

RF4 – Feedback	
Título: TBMJIRA-107 Visualización de una entrega	Estimación: 10 SP
<p>User Story:</p> <p>Como usuario responsable de un curso Quiero ver la entrega realizada por otro usuario a un ejercicio de mi curso Para para poder revisarla y darle feedback.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-107-AC-1: Visualización de una entrega exitosa</u> Dada una entrega realizada por otro usuario a un ejercicio de un curso Cuando el usuario responsable del curso ingresa al ejercicio Y hace clic en ver entrega Entonces se visualiza lo entregado.</p> <p><u>TBMJIRA-107-AC-2: Visualización de una entrega fallida sin conexión a Internet</u> Dada una entrega realizada por otro usuario a un ejercicio de un curso Cuando el usuario responsable del curso ingresa al ejercicio Entonces el sistema informa que no se pudo cargar el contenido.</p>	

Tabla 57 - Detalle de US TBMJIRA-107

Tareas contenidas en TBMJIRA-107 Visualización de una entrega		
ID	Nombre	SP
TBMJIRA-108	[FE] Creación de componente	5
TBMJIRA-109	[FE] Conexión con el backend	2
TBMJIRA-110	[BE] Creación del endpoint	3

Tabla 58 - Tareas dentro de TBMJIRA-107

RF4 – Feedback	
Título: TBMJIRA-113 Agregar comentarios a una entrega	Estimación: 18 SP
<p>User Story:</p> <p>Como usuario responsable de un curso Quiero agregar comentarios a una entrega realizada por otro usuario a un ejercicio de mi curso Para darle feedback.</p>	

Criterios de aceptación (escenarios):TBMJIRA-113-AC-1: Adición de comentarios a una entrega exitosa**Dado** una entrega realizada por otro usuario a un ejercicio de un curso**Cuando** el usuario responsable del curso ingresa al ejercicio**Y** rellena el campo para los comentarios**Y** presiona el botón de enviar feedback**Entonces** los comentarios son agregados a la entrega**Y** el estado del ejercicio pasa a ser entregado y revisado.TBMJIRA-113-AC-2: Adición de comentarios a una entrega fallida sin conexión a Internet**Dado** una entrega realizada por otro usuario a un ejercicio de un curso**Cuando** el usuario responsable del curso ingresa al ejercicio**Entonces** el sistema informa que no se pudo cargar el contenido.

Tabla 59 - Detalle de US TBMJIRA-113

Tareas contenidas en TBMJIRA-113 Agregar comentarios a una entrega		
ID	Nombre	SP
TBMJIRA-114	[FE] Creación de componente	8
TBMJIRA-115	[BE] Creación del endpoint	5
TBMJIRA-116	[FE] Conexión con el backend	5

Tabla 60 - Tareas dentro de TBMJIRA-113

RF4 – Feedback	
Título: TBMJIRA-117 Visualización de comentarios en un ejercicio	Estimación: 8 SP
User Story: Como usuario Quiero ver los comentarios realizados a mi entrega Para poder leer el feedback y corregir posibles errores.	
Criterios de aceptación (escenarios): <u>TBMJIRA-117-AC-1: Visualización de comentarios en un ejercicio exitosa</u> Dado un usuario que realizó una entrega que fue revisada Cuando ingresa al ejercicio Entonces se muestran los comentarios realizados por el responsable del curso. <u>TBMJIRA-117-AC-2: Visualización de comentarios en un ejercicio fallida sin conexión a Internet</u> Dado un usuario que realizó una entrega que fue revisada Cuando ingresa al ejercicio Entonces el sistema informa que no se pudo cargar el contenido.	

Tabla 61 - Detalle de US TBMJIRA-117

Tareas contenidas en TBMJIRA-117 Visualización de comentarios en un ejercicio		
ID	Nombre	SP
TBMJIRA-118	[BE] Creación del endpoint	2
TBMJIRA-119	[FE] Creación de componente	5
TBMJIRA-120	[FE] Conexión con el backend	1

Tabla 62 - Tareas dentro de TBMJIRA-117

RF5 – Favoritos	
Título: TBMJIRA-126 Búsqueda de un curso	Estimación: 13 SP
<p>User Story:</p> <p>Como usuario Quiero buscar un curso en particular Para encontrarlo más fácil y rápidamente.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-126-AC-1: Búsqueda de un curso exitosa</u> Dada una lista de cursos Cuando se ingresa un texto en el buscador Entonces se muestran resultados que coinciden con el criterio de búsqueda.</p> <p><u>TBMJIRA-126-AC-2: Búsqueda de un curso exitosa sin coincidencias</u> Dada una lista de cursos Cuando se ingresa un texto en el buscador Y no hay cursos que coincidan con el criterio de búsqueda Entonces el sistema informa que no hay resultados para la búsqueda.</p> <p><u>TBMJIRA-126-AC-3: Búsqueda de un curso fallida sin conexión a Internet</u> Dada una lista de cursos Cuando se ingresa un texto en el buscador Entonces el sistema informa que no se pudo cargar el contenido.</p>	

Tabla 63 - Detalle de US TBMJIRA-126

Tareas contenidas en TBMJIRA-126 Búsqueda de un curso		
ID	Nombre	SP
TBMJIRA-127	[CD] Configuración de OpenSearch para búsqueda por criterio	5
TBMJIRA-128	[BE] Conexión con OpenSearch	2
TBMJIRA-129	[FE] Creación de componente	2
TBMJIRA-130	[BE] Creación del endpoint	2
TBMJIRA-131	[FE] Conexión con el backend	2

Tabla 64 - Tareas dentro de TBMJIRA-126

RF5 – Favoritos	
Título: TBMJIRA-132 Selección de un curso como favorito	Estimación: 8 SP
<p>User Story:</p> <p>Como usuario Quiero agregar un curso a favoritos Para acceder a él más rápidamente y poder verlo más tarde.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-132-AC-1: Selección de un curso como favorito exitosa</u> Dada una lista de cursos Cuando se ingresa a un curso Y se hace clic en el botón para agregar a favoritos Entonces el curso se añade a la lista de favoritos del usuario Y el botón cambia su texto a “remove favorito”.</p> <p><u>TBMJIRA-132-AC-2: Selección de un curso como favorito fallida sin conexión a Internet</u> Dada una lista de cursos Cuando se ingresa a un curso Entonces el sistema informa que no se pudo cargar el contenido.</p>	

Tabla 65 - Detalle de US TBMJIRA-132

Tareas contenidas en TBMJIRA-132 Selección de un curso como favorito		
ID	Nombre	SP
TBMJIRA-133	[BE] Creación del endpoints necesarios	5
TBMJIRA-134	[FE] Creación de componente	2
TBMJIRA-135	[FE] Conexión con el backend	1

Tabla 66 - Tareas dentro de TBMJIRA-132

RF5 – Favoritos	
Título: TBMJIRA-136 Listado de cursos favoritos	Estimación: 8 SP
<p>User Story:</p> <p>Como usuario Quiero listar los cursos que agregué a favoritos Para acceder a ellos más rápidamente.</p>	

Criterios de aceptación (escenarios):*TBMJIRA-136-AC-1: Listado de cursos favoritos exitoso***Dada** una lista de cursos**Cuando** se hace clic en la opción de menú correspondiente a favoritos**Entonces** se listan los cursos agregados a favoritos del usuario.*TBMJIRA-136-AC-2: Listado de cursos favoritos exitoso sin resultados***Dada** una lista de cursos**Cuando** se hace clic en la opción de menú correspondiente a favoritos**Y** no se han añadido cursos aún**Entonces** el sistema informa que no hay resultados.*TBMJIRA-136-AC-3: Listado de cursos favoritos fallido sin conexión a Internet***Dada** una lista de cursos**Cuando** se hace clic en la opción de menú correspondiente a favoritos**Entonces** el sistema informa que no se pudo cargar el contenido.

Tabla 67 - Detalle de US TBMJIRA-136

Tareas contenidas en TBMJIRA-136 Listado de cursos favoritos		
ID	Nombre	SP
TBMJIRA-137	[BE] Creación del endpoint	5
TBMJIRA-138	[FE] Creación de componente	2
TBMJIRA-139	[FE] Conexión con el backend	1

Tabla 68 - Tareas dentro de TBMJIRA-136

RF6 – Creación de curso	
Título: TBMJIRA-147 Creación de curso nuevo (admin)	Estimación: 21 SP
User Story: Como usuario administrador Quiero crear un curso en la plataforma Para compartir contenido útil a los usuarios.	

Criterios de aceptación (escenarios):*TBMJIRA-147-AC-1: Creación de curso exitosa***Dado** un usuario administrador**Cuando** hace clic en la opción del menú correspondiente a crear un curso**E** ingresa título, descripción, su email**Y** hace clic en siguiente**Y** agrega módulos conteniendo lecciones válidas**Y** hace clic en siguiente**Entonces** se muestra una vista previa del curso**Y** se muestra un botón para publicarlo.*TBMJIRA-147-AC-2: Creación de curso fallida datos inválidos***Dado** un usuario administrador**Cuando** hace clic en la opción del menú correspondiente a crear un curso**E** ingresa algún dato inválido**Entonces** el sistema informa el error correspondiente**Y** no permite avanzar al siguiente paso.*TBMJIRA-147-AC-3: Creación de curso fallida sin conexión a Internet***Dado** un usuario administrador**Cuando** hace clic en la opción del menú correspondiente a crear un curso**Entonces** el sistema informa que no se pudo cargar el contenido.

Tabla 69 - Detalle de US TBMJIRA-147

Tareas contenidas en TBMJIRA-147 Creación de curso nuevo (admin)		
ID	Nombre	SP
TBMJIRA-148	[BE] Creación del endpoint	13
TBMJIRA-149	[FE] Creación de componente	5
TBMJIRA-150	[FE] Conexión con el backend	3

Tabla 70 - Tareas dentro de TBMJIRA-147

RF6 – Creación de curso	
Título: TBMJIRA-151 Listado de cursos creados	Estimación: 3 SP
User Story: Como usuario responsable de un curso Quiero ver un listado de mis cursos Para asegurarme que su contenido sea correcto.	

Criterios de aceptación (escenarios):*TBMJIRA-151-AC-1: Listado de cursos creados exitoso***Dado** un usuario responsable de un curso**Cuando** hace clic en la opción del menú correspondiente a inicio**Entonces** se muestra un listado de sus cursos.*TBMJIRA-151-AC-2: Listado de cursos creados exitoso sin resultados***Dado** un usuario responsable de un curso**Cuando** hace clic en la opción del menú correspondiente a inicio**Entonces** el sistema informa que no hay resultados.*TBMJIRA-151-AC-3: Listado de cursos creados fallido sin conexión a Internet***Dado** un usuario responsable de un curso**Cuando** hace clic en la opción del menú correspondiente a inicio**Entonces** el sistema informa que no se pudo cargar el contenido.

Tabla 71 - Detalle de US TBMJIRA-151

Tareas contenidas en TBMJIRA-151 Listado de cursos creados		
ID	Nombre	SP
TBMJIRA-152	[BE] Creación del endpoint	1
TBMJIRA-153	[FE] Creación de componente	1
TBMJIRA-154	[FE] Conexión con el backend	1

Tabla 72 - Tareas dentro de TBMJIRA-151

RF6 – Creación de curso	
Título: TBMJIRA-155 Modificación de curso (admin)	Estimación: 5 SP
User Story: Como usuario administrador Quiero modificar un curso Para editar su contenido.	
Criterios de aceptación (escenarios): <u><i>TBMJIRA-155-AC-1: Modificación de curso exitosa</i></u> Dado un usuario administrador Cuando se listan todos los cursos Y hace clic en el botón modificar curso para uno en particular Entonces el curso pasa a estar en estado inactivo Y se muestran los datos actuales del curso Y se dejan modificar.	
<u><i>TBMJIRA-155-AC-2: Modificación de curso fallida sin conexión a Internet</i></u> Dado un usuario administrador Cuando se listan todos los cursos Entonces el sistema informa que no se pudo cargar el contenido.	

Tabla 73 - Detalle de US TBMJIRA-155

Tareas contenidas en TBMJIRA-155 Modificación de curso (admin)		
ID	Nombre	SP
TBMJIRA-156	[BE] Creación del endpoint	2
TBMJIRA-157	[FE] Creación de componente	1
TBMJIRA-158	[FE] Conexión con el backend	2

Tabla 74 - Tareas dentro de TBMJIRA-155

RF6 – Creación de curso	
Título: TBMJIRA-159 Solicitud de creación de curso	Estimación: 13 SP
<p>User Story:</p> <p>Como usuario Quiero solicitar la creación de un curso en la plataforma Para compartir mis conocimientos con el resto de los usuarios.</p>	
<p>Criterios de aceptación (escenarios):</p> <p><u>TBMJIRA-159-AC-1: Solicitud de creación de curso exitosa</u> Dado un usuario Cuando hace clic en la opción del menú correspondiente a solicitar crear un curso Entonces se muestra un form para completar.</p> <p><u>TBMJIRA-159-AC-2: Solicitud de creación de curso fallida sin conexión a Internet</u> Dado un usuario Cuando hace clic en la opción del menú correspondiente a solicitar crear un curso Entonces el sistema informa que no se pudo cargar el contenido.</p>	

Tabla 75 - Detalle de US TBMJIRA-159

Tareas contenidas en TBMJIRA-159 Solicitud de creación de curso		
ID	Nombre	SP
TBMJIRA-160	[BE] Creación del endpoint	5
TBMJIRA-161	[FE] Creación de componente	5
TBMJIRA-162	[FE] Conexión con el backend	3

Tabla 76 - Tareas dentro de TBMJIRA-159

RF6 – Creación de curso	
Título: TBMJIRA-163 Listado de solicitudes de creación de curso (admin)	Estimación: 8 SP
<p>User Story:</p> <p>Como usuario administrador Quiero ver las solicitudes para crear un curso Para revisarlas y ponerme en contacto con los usuarios que las enviaron.</p>	

Criterios de aceptación (escenarios):

TBMJIRA-163-AC-1: Listado de solicitudes de creación de curso exitosa

Dado un usuario administrador

Cuando hace clic en la opción de menú correspondiente a ver las solicitudes para creación de un curso enviadas por usuarios

Entonces se listan todas las solicitudes recibidas.

TBMJIRA-163-AC-2: Listado de solicitudes de creación de curso fallido sin conexión a Internet

Dado un usuario administrador

Cuando hace clic en la opción de menú correspondiente a ver las solicitudes para creación de un curso enviadas por usuarios

Entonces el sistema informa que no se pudo cargar el contenido.

Tabla 77 - Detalle de US TBMJIRA-163

Tareas contenidas en TBMJIRA-163 Listado de solicitudes de creación de curso (admin)		
ID	Nombre	SP
TBMJIRA-164	[BE] Creación del endpoint	2
TBMJIRA-165	[FE] Creación de componente	5
TBMJIRA-166	[FE] Conexión con el backend	1

Tabla 78 - Tareas dentro de TBMJIRA-163

13.7. Anexo 7: Catálogo de elementos del diagrama general

A continuación, se detallan los elementos presentados en el diagrama de la sección [7.1.2. Visión general](#), basándonos a su vez en la documentación oficial para explicar los mismos. [9]

- **AWS Amplify Hosting:** Servicio de AWS que permite el alojamiento, integración continua (CI) y despliegue continuo (CD) de aplicaciones web estáticas, de forma rápida y segura, pudiendo manejar variables de entorno y distintos entornos, utilizado para deployar nuestra aplicación web.
- **AWS API Gateway:** Servicio de AWS encargado de distribuir las request http transformadas a eventos soportados por la función api handler, contienen la lógica para manejar dichas requests.
- **AWS CloudFormation:** Servicio de AWS que permite modelar y configurar los recursos de AWS de manera fácil y ágil, facilitando el despliegue, conexión entre

recursos y manejo de distintos ambientes (producción, desarrollo, test, entre otros), sin contar la Webapp. En este caso manejado mediante Serverless Framework.

- **AWS CloudFront:** Servicio de AWS que nos permite entregar contenido, ofreciendo un alto rendimiento y seguridad, integrándose con servicios de contenido como puede ser AWS S3 y Lambda@Edge. En nuestro caso, utilizado por nuestra aplicación web.
- **AWS CloudWatch:** Servicio de AWS que nos permite el almacenamiento y registro de logs de distintos tipos de servicios de AWS. En nuestro caso, utilizado para mantener un registro de las actividades ocurridas en las lambdas.
- **AWS Cognito:** Servicio de AWS que ofrece autenticación, autorización y administración de usuarios para nuestra plataforma.
- **AWS DynamoDB:** Servicio de AWS que nos permite el almacenamiento y manejo de nuestras bases de datos NoSQL, del tipo clave-valor.
- **AWS EventBridge:** Servicio de AWS que utiliza eventos para la comunicación asincrónica entre distintos componentes de nuestra aplicación, utilizado para la creación de cursos.
- **AWS Lambda@Edge (auth):** Servicio de AWS, utilizado para ejecutar código asociado a una instancia de AWS CloudFront sin necesidad de administrar infraestructura en distintas partes del mundo, mejorando la latencia y el rendimiento. En este caso, utilizada para autorizar el acceso al contenido solicitado en AWS S3, mediante AWS CloudFront, validando la existencia del usuario por medio de AWS Cognito y que el origen de la request sea uno de los permitidos.
- **AWS Lambda:** Servicio de AWS que permite ejecutar código sin necesidad de aprovisionar ni administrar servidores, por medio de funciones.
- **AWS Lambda (auth):** Función que cumple el rol de autorizar o no el acceso a un recurso de nuestra api (api handler), validando la existencia del usuario por medio de AWS Cognito y que el origen de la request sea uno de los permitidos.

- **AWS Lambda (api handler):** Función que contiene una REST API, la cual, recibe los eventos generados por AWS API Gateway y los dirige a cada recurso en específico dependiendo del contenido de la request HTTP.
- **AWS Lambda (event handler):** Función encargada de recibir y manejar ciertos eventos que llegan a AWS EventBridge, realizando acciones dependiendo su tipo.
- **AWS MediaConvert:** Servicio de AWS encargado de transcodificar video basado en archivos con funciones de transmisión. En nuestro caso, particionar los videos en varias partes y calidades para poder reproducir videos mediante AWS CloudFront y un reproductor de video, que permita este tipo de carga en varias definiciones. Permite la codificación de varios videos a la vez y su priorización es en base al orden de llegada.
- **AWS OpenSearch:** Servicio de AWS que permite el administrar, implementar, operar y escalar clústeres del motor de búsqueda de OpenSearch en la nube. En nuestro caso utilizado para la búsqueda avanzada de cursos.
- **Serverless:** Nos permite crear y ejecutar nuestra aplicación con rapidez, englobando todos los recursos y sus conexiones en un solo archivo, utilizando servidores de terceros. En nuestro caso, despliega para cada ambiente todos los servicios detallados en el diagrama con la ayuda de AWS CloudFront menos la aplicación web. [10]
- **Webapp React.js:** Aplicación web que sigue los estándares de una SPA, construida con React y desplegada mediante el uso de AWS Amplify Hosting.

13.8. Anexo 8: Vistas de componentes y conectores

En las siguientes vistas se demostrará como se integran en tiempo de ejecución nuestras funciones de AWS Lambda con los distintos servicios que utilizamos. Teniendo en cuenta que un componente es algo que se manifiesta en tiempo de ejecución, podemos considerar a las funciones en AWS como componentes. [27]

- **Inicial**

En la siguiente imagen (Figura 73) se puede visualizar el diagrama inicial de componentes y conectores (C&C).

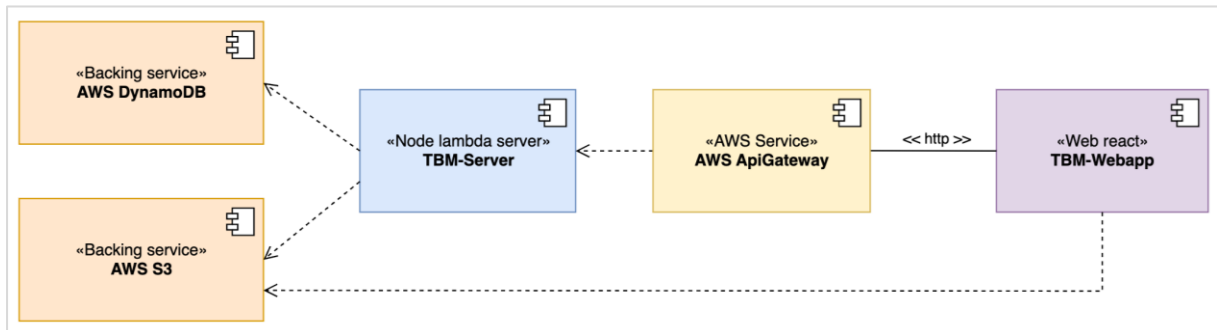


Figura 73 - Diagrama de C&C inicial

En el mismo se puede visualizar un Frontend, utilizando React.js. Luego, una función del tipo AWS Lambda que contiene un RESTfull API a la cual se accede mediante AWS API Gateway que se encarga de transformar las request HTTP a eventos que entiende nuestra función. Esta función accede a los datos de los recursos almacenados utilizando AWS DynamoDB y a los archivos mediante AWS S3. Es importante destacar que el Frontend accede directamente a AWS S3 para poder mostrar los archivos.

Las líneas punteadas indican que las comunicaciones son internas al tratarse de servicios de AWS en la misma región. No sucede lo mismo con la conexión entre AWS API Gateway y TBM-Webapp, la cual se da mediante el uso del protocolo HTTP.

- **Vista genérica final**

Esta vista representa el resultado final obtenido luego de varias iteraciones y validaciones (Figura 74).

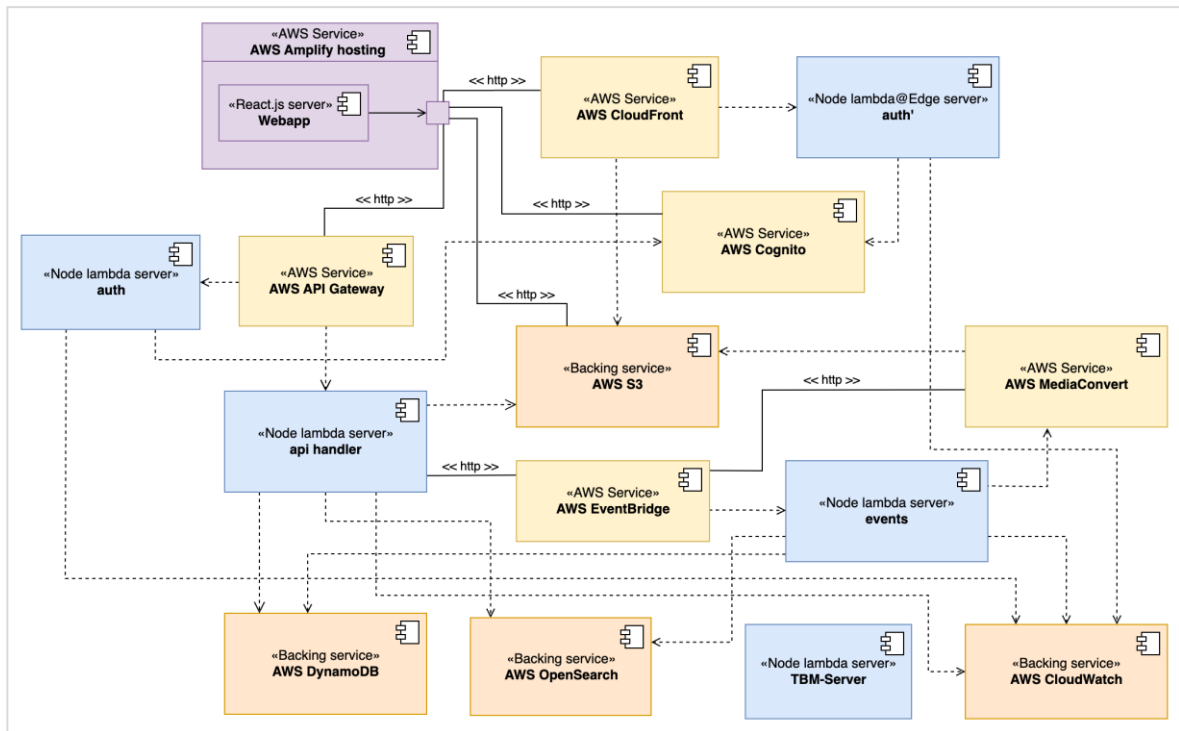


Figura 74 - Diagrama de C&C final

Para esta vista, no se describieron en profundidad los elementos y decisiones de diseño debido a que son similares a lo explicado en la sección [7.1.2. Visión general](#), con la diferencia de que este diagrama sigue las estructuras y conectores de una vista de componentes y conectores.

Las líneas punteadas refieren a componentes que son utilizados dentro de la misma región de AWS, lo que hace la interacción entre ellos más rápida. En el caso de AWS MediaConvert y EventBridge para el mecanismo de detección de eventos o conversión de nuevos videos funciona para distintas regiones por lo no se tienen en cuenta este tipo de conexiones como internas.

13.9. Anexo 9: Estructura de datos en Redux

Se crearon reducers que acompañan las funcionalidades más importantes de nuestra aplicación web, cada uno de ellos contiene información útil, que es observada por los componentes que forman parte de cada una de estas funcionalidades. A continuación, se describirán los reducers utilizados, comentando el tipo de información que almacenan y a que funcionalidad están enlazados.

- **courseCreator:** Contiene toda la información que se utiliza mientras se crea o actualiza un curso, a su vez, también tiene la lista de solicitudes de creación de un curso. Por poner ejemplos, contiene en que paso del flujo nos encontramos, si sucedió algún error, hasta el curso que se está modificando, entre otros.
- **courses:** Podría decirse que este reducer es el más importante de la plataforma, ya que contiene información que es utilizada en la búsqueda de cursos, la vista en detalle de uno en particular, la vista de favoritos y la visualización de videos o documentos. Por poner ejemplos, contiene el curso seleccionado, la lista de cursos con variables relacionadas al paginado y la lista de favoritos, entre otros.
- **exercises:** Este reducer contiene toda la información relacionada a un ejercicio y al listado de estos, tanto para el docente los ejercicios pendientes de feedback, como para el estudiante los ejercicios entregados y su estado. Por poner ejemplos, contiene el ejercicio seleccionado, listado de ejercicios de docente y estudiante, entre otros.
- **site:** Este reducer es común a toda la plataforma, y se utiliza para saber si se realizó correctamente la carga de datos inicial, en caso de algún fallo en la carga de la plataforma al cargarla por primera vez, como puede ser, no se pudo verificar si se está logueado o no, esa información estará en este reducer.
- **users:** Este reducer contiene la información relacionada al usuario logueado, como es sus datos y notificaciones. Se utiliza en casi todas las partes de la plataforma, ya que al contener la información del usuario logueado, esta es necesaria para realizar acciones en su nombre.
- **videos:** Este reducer cumple el rol de almacenar el estado de la reproducción de videos, siendo utilizado para videos de uno a tres ángulos de cámara y en particular, para el despliegue del quiz en el video de realidad virtual, almacenando las preguntas que fueron apareciendo y las que no.

13.10. Anexo 10: Endpoints

A continuación, se listarán los endpoints de nuestra API con información general, separados por recurso, indicando su verbo, dirección, si admiten query strings, requieren un body y su objetivo. Todos los endpoints de nuestro sistema requieren autorización, realizada con la función en AWS Lambda de nombre auth.

Es importante destacar que este listado no busca ser una especificación completa de la API si no mostrar de forma general la estructura y diseño de la misma.

Formato: [Verbo] {URL}/dirección?unQueryString={string_value}... [Body si tiene] : objetivo.

▪ Users

- [GET] {URL}/users/me : Obtiene la información del usuario actual.
- [GET] {URL}/users/me/notifications : Obtiene las notificaciones del usuario actual.
- [POST] {URL}/users [Body]: Marca las notificaciones del usuario actual como vistas.
- [POST] {URL}/users/me/notifications/seen : Marca las notificaciones del usuario actual como vistas.
- [GET] {URL}/users/me/notifications/seen : Obtiene si las notificaciones del usuario actual han sido vistas.
- [GET] {URL}/users/me/favorites : Obtiene los cursos favoritos del usuario actual.
- [GET] {URL}/users/me/favorites/:courseId : Obtiene si un curso es favorito del usuario actual.
- [POST] {URL}/users/me/favorites/:courseId [Body]: Agrega un curso a la lista de favoritos del usuario actual.

- [DELETE] {URL}/users/me/favorites/:courseId : Elimina un curso de la lista de favoritos del usuario actual.
- [GET] {URL}/users/:id : Obtiene la información de un usuario específico según su ID.

▪ Ejercicios

- [GET] {URL}/exercises/student?lastEvaluatedKey={string_value}&take={string_value} : Obtiene los ejercicios del estudiante actual con paginación.
- [GET] {URL}/exercises/teacher?lastEvaluatedKey={string_value}&take={string_value} : Obtiene los ejercicios del profesor actual sin comentarios con paginación.
- [POST] {URL}/exercises [Body] : Crea un nuevo ejercicio.
- [GET] {URL}/exercises/:id : Obtiene la información de un ejercicio específico según su ID.
- [GET] {URL}/exercises/me/:courseId/lesson/:lessonId : Obtiene el ejercicio correspondiente al estudiante actual para una lección específica en un curso.
- [POST] {URL}/exercises/:id/feedback [Body] : Agrega un comentario sobre un ejercicio específico.

▪ Archivos

- [POST] {URL}/files [Body]: Crea una URL firmada para subir un archivo.
- [GET] {URL}/files/:id?lastEvaluatedKey=string&take=integer : Obtiene la lista de objetos de un archivo específico según su ID. Los parámetros opcionales lastEvaluatedKey y take se utilizan para la paginación.
- [DELETE] {URL}/files/:id : Elimina un objeto de un archivo específico según su ID.

▪ Cursos

- [GET]
{URL}/courses?criteria={string_value}&take={integer}&from={integer}:
Obtener todos los cursos.
- [GET] {URL}/courses/:id : Obtener un curso por ID.
- [POST] {URL}/courses [Body]: Crear o modificar un curso.
- [POST] {URL}/courses/requests [Body]: Crear una solicitud de creación de un nuevo curso.
- [POST] {URL}/courses/:id/publish [Body]: Publicar un curso.
- [GET]
{URL}/courses/:id/reviews?lastEvaluatedKey={lastEvaluatedKey}&take={integer} : Obtener las reviews de un usuario de un curso con paginación.
- [POST] {URL}/courses/:id/reviews [Body]: Crear una nueva reseña para un curso.
- [PUT] {URL}/courses/:id/reviews [Body]: Actualizar una review existente de un curso.
- [POST] {URL}/courses/:id/questions [Body]: Añadir una pregunta a un curso.
- [GET]
{URL}/courses/:id/questions?lastEvaluatedKey={lastEvaluatedKey}&take={integer} : Obtener las preguntas de un curso con paginación
- [POST] {URL}/courses/:id/questions/:questionId/answers [Body]: Añadir una respuesta a una pregunta de un curso.
- [GET]
{URL}/courses/requests?lastEvaluatedKey={lastEvaluatedKey}&take={integer} : Obtener las solicitudes de creación de curso con paginación.

13.11. Anexo 11: Tablas y Colecciones

A continuación, se mostrará el diagrama de relaciones de nuestras tablas en AWS DynamoDB. Dado que esta forma de diagramar nuestras tablas no es una de las conocidas o utilizadas generalmente, es necesario realizar ciertas aclaraciones estructurales (Figura 75).

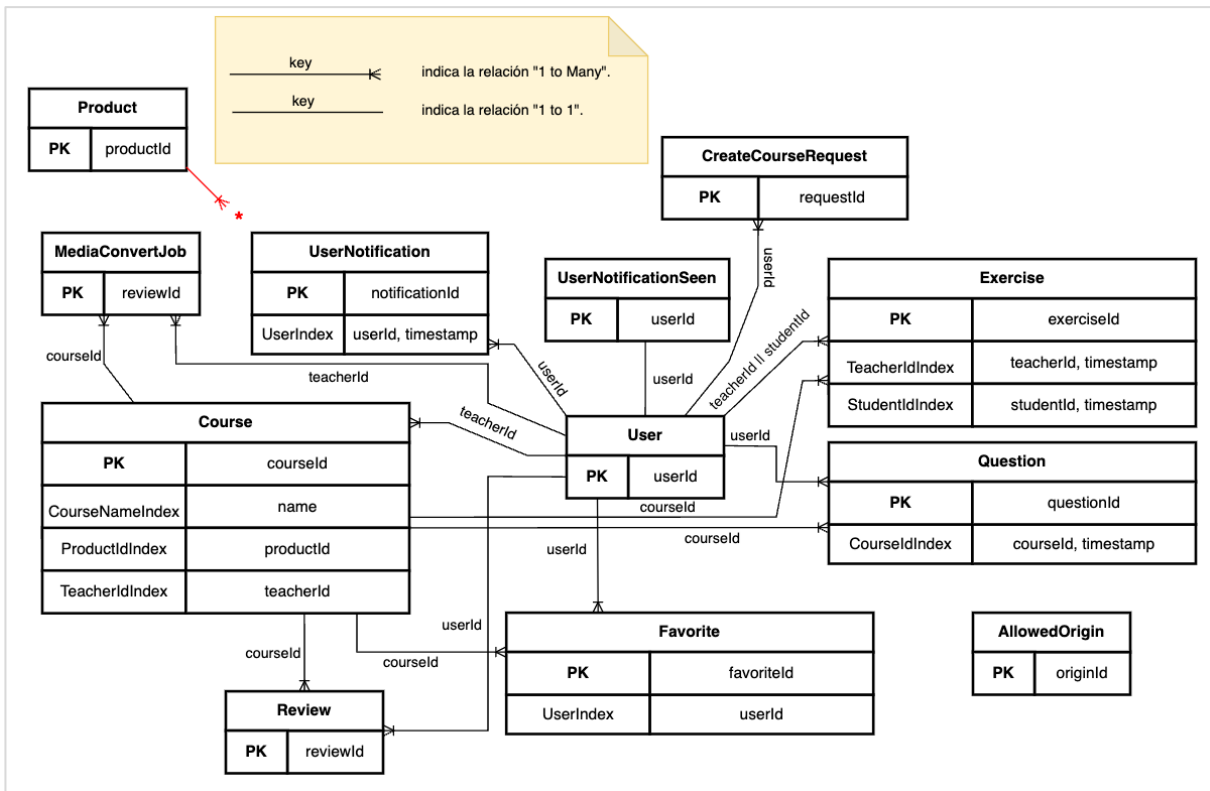


Figura 75 - Tablas de AWS DynamoDB

En el diagrama anterior se puede visualizar las tablas con sus claves primarias y sus respectivos índices.

Es importante destacar que la relación más recurrente es similar al "1 to Many" utilizado en el modelado entidad relación. Por ejemplo, entre la tabla Users y Questions se visualiza que un usuario en particular puede tener indefinidas preguntas que contengan su clave primaria. Cada relación indica el nombre por el que guardará la referencia a la clave primaria en la otra tabla, siguiendo con el mismo ejemplo, cada pregunta contendrá un atributo userId haciendo referencia al identificador de un usuario en particular. En el caso de la tabla ejercicios, tiene dos identificadores que representan a distintos usuarios dentro del ejercicio, uno el profesor corrector y el otro el estudiante.

El caso en particular de las tablas User y UserNotificationsSeen representa la relación equivalente al “1 a 1” en un modelado entidad relación.

El caso de la Tabla Products mantiene la relación “1 to Many” con todas las tablas del sistema.

A continuación, se listan las tablas y sus responsabilidades (Tabla 79).

Tabla	Responsabilidad
AllowedOrigin	Almacena los orígenes permitidos (URL) para los determinados productos del sistema.
Course	Es la encargada de almacenar la información relacionada a un curso, módulos y lecciones, además de otros datos claves para su visualización completa.
CourseSearch	Tabla de AWS OpenSearch, encargada de mantener un subconjunto de datos de un curso necesarios para su búsqueda.
CreateCourseRequest	Es la encargada de almacenar las solicitudes de creación de cursos.
Exercise	Almacena la información de un ejercicio entregado a un estudiante y su posterior feedback recibido.
Favorite	Almacena los cursos favoritos de un usuario
MediaConvertJob	Almacena el estado de conversión de un video, el cual puede ser en Progreso, Error o Completado.
Product	Almacena las variables de determinado producto, como puede ser, templates de AWS MediaConvert o URL para comunicarse con la API.
Question	Encargada de almacenar las preguntas y respuestas de un curso.
Review	Almacena la puntuación de un usuario a un curso.
UserNotificationSeen	Almacena únicamente si un usuario en particular vio o no sus notificaciones.
UserNotification	Almacena las notificaciones de los usuarios.
Users	Almacena la información del usuario logueado.

Tabla 79 - Tablas y sus correspondientes responsabilidades

A continuación, se mostrará la estructura de las colecciones utilizadas en nuestra plataforma, incluida la tabla de AWS OpenSearch.

En particular destacamos la utilizada para la tabla de cursos, siendo esta la principal estructura de nuestra plataforma. Como dato adicional, dicha estructura está relacionada al lenguaje TypeScript. A su vez, sigue el patrón de modelado de tablas recomendado “Patrón del revisor” [1], debido a que los módulos y lecciones dentro de estos, no son compartidos por otros cursos. Ver figura que tiene la colección de Course (Figura 76).

```
Course:
{
  courseId: string;
  name: string;
  description: string;
  descriptionHTML: string;
  teacherId: string;
  totalDuration: number;
  picture: string;
  productId: string;
  status: number;
  amountOfReviews: number;
  averageReviews: number;
  totalReviews: number;
  modules:
  [
    moduleId: string;
    name: string;
    index: number;
    lessons:
    [
      lessonId: string;
      name: string;
      duration: number;
      type: string;
      link: string | string[];
      linkPath: string | string[];
      state: string;
      cameraAngles: number;
      index: number;
      quiz:
      [
        type: string;
        question: string;
        options: string[];
        answer: string;
        explanation: string;
        time: number;
      ];
    ];
  ];
};
}
```

Figura 76 - Tabla de Curso

```
UserNotificationSeen:
{
  userId: string;
  areNotificationsSeen: boolean;
}
```

Figura 77 - Tabla que indica si las notificaciones fueron vistas por el usuario

```
User:
{
  userId: string;
  email: string;
  productId: string;
  firstName: string;
  lastName: string;
  picture: string;
  isAdmin: boolean;
}
```

Figura 78 - Tabla de usuario

```
CreateCourseRequest:
{
  requestId: string;
  userId: string;
  description: string;
  linkPath: string;
  optionalLink: string;
}
```

Figura 79 - Tabla de solicitudes de creación de un curso

```
CourseSearch:
{
  courseId: string;
  name: string;
  description: string;
  descriptionHTML: string;
  teacherId: string;
  totalDuration: number;
  picture: string;
  productId: string;
  status: number;
  teacherName:string;
}
```

Figura 80 - Tabla de Cursos para AWS OpenSearch

```
MediaConvertJob:
{
  jobId: string;
  courseId: string;
  productId: string;
  moduleIndex: number;
  lessonIndex: number;
  link: string;
  linkPath: string;
  jobStatus: string;
  jobType: string;
}
```

Figura 81 - Tabla de trabajos realizados por AWS MediaConvert

```
Favorite:
{
  favoriteId: string;
  courseId: string;
  userId: string;
  productId: string;
}
```

Figura 82 - Tabla de cursos favoritos de un usuario

```
Question:
{
  questionId: string;
  studentId: string;
  teacherId: string;
  timestamp: number;
  courseId: string;
  questionText: string;
  answer: string | null;
  productId: string;
}
```

Figura 83 - Tabla de preguntas de un curso

```
Product:
{
  productId: string;
  data: Object;
}
```

Figura 84 - Tabla de productos

```
UserNotification:
{
  notificationId: string;
  userId: string;
  type: string;
  data: Object;
  timestamp: number;
  productId: string;
}
```

Figura 85 - Tabla de notificaciones de un usuario

```
Exercise:
{
  exerciseld: string;
  courseId: string;
  lessonId: string;
  teacherId: string;
  studentId: string;
  productId: string;
  teacherEmail: string;
  studentEmail: string;
  feedback:
  {
    comments: string;
    deliveredFeedback: string;
  };
  deliveredExercise: string;
  status: number;
  timestamp: number;
}
```

Figura 86 - Tabla de ejercicios de un usuario

```
AllowedOrigin:
{
  originId: string;
  productId: string;
}
```

Figura 87 - Tabla de orígenes permitidos por producto

```
Review:
{
  reviewId: string;
  courseId: string;
  userId: string;
  productId: string;
  stars: number;
}
```

Figura 88 - Tabla de reseñas

Referencias bibliográficas

[1] Redis, “NoSQL Data Modeling with Redis”, 2022 [Online]. Disponible: <https://redis.com/blog/nosql-data-modeling/>

13.12. Anexo 12: Crear un nuevo producto

A continuación, se describirá los pasos a seguir para crear un nuevo producto. Primero que nada, es necesario setear para crear un nuevo producto su clave, la cual debe ser en minúscula y sin espacios, ejemplo “Stiler”.

Pasos a seguir para el Backend:

1. Configurar el user-pool en AWS Cognito para cada ambiente con el nuevo producto.
2. Agregar los dominios permitidos en la tabla de allowed-origins-table de cada ambiente correspondiente al nuevo producto, a modo de ejemplo actual, allowed-origins-table-dev y allowed-origins-table-prod.
3. Luego agregar un nuevo record para cada ambiente en la tabla products-table-{ambiente}, que especifique la configuración del producto (ver los records ya existentes). Acá, se puede utilizar un template distinto o el genérico de AWS MediaConvert, buckets distintos de AWS S3, pero por defecto se recomienda utilizar los mismos buckets a modo de optimizar recursos.

Pasos a seguir para el Frontend:

1. Ir a AWS-Amplify y crear un nuevo producto y deployar la rama relacionada a cada ambiente.
2. Para cada nuevo ambiente, configurar las variables de entorno correspondiente a este nuevo producto.
3. Validar que la integración y despliegue continuo este activado.

Con estos pasos, ya se tiene configurado un nuevo producto. Es importante destacar, que en etapas posteriores al proyecto y con un cliente se puede personalizar más funcionalidades, como podría ser activar o desactivar los videos 360, por ejemplo.

13.13. Anexo 13: Diagrama de paquete de capas

A continuación, se puede visualizar el diagrama de paquetes, el cual, utiliza inyección de dependencias junto con el principio de inversión de dependencias de los principios SOLID para la conexión entre las capas, ya que se pueden ver las interfaces bien definidas que conectan a las capas (Figura 89).

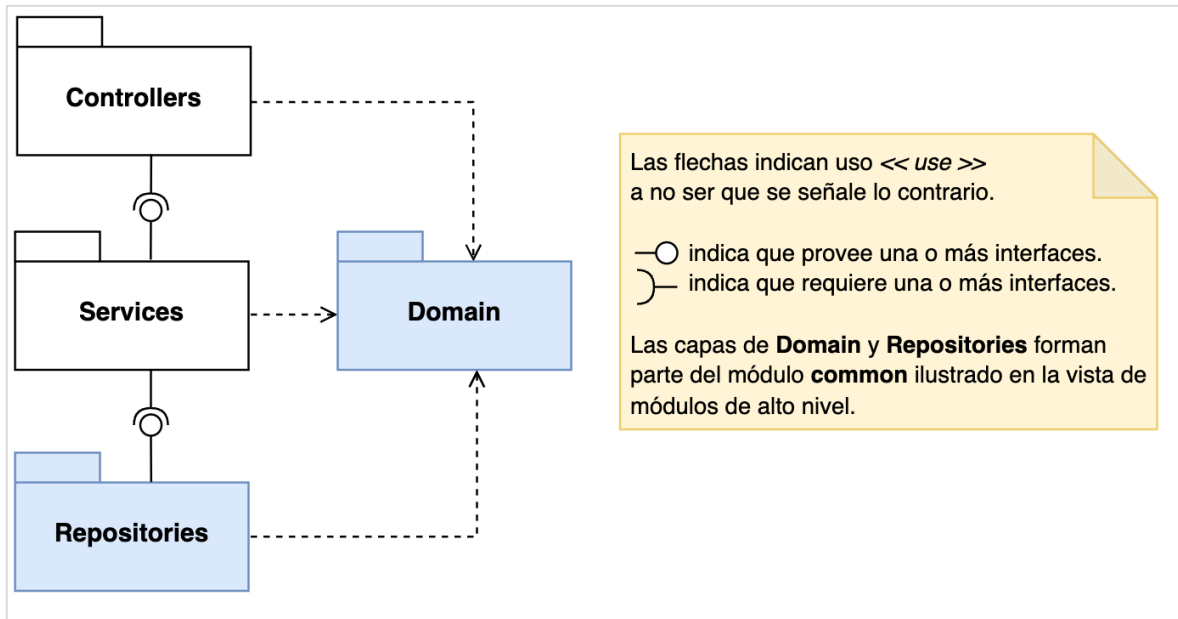


Figura 89 - Diagrama de paquetes

13.14. Anexo 14: Versiones de la planificación anual

A continuación, mostramos las diversas versiones del plan anual que realizamos, reflejando cambios que ocurrieron durante la primera etapa de gestión del proyecto.

▪ **Planeación anual 2022 – 2023 versión 1**



Figura 90 - Planeación anual versión 1

▪ **Planeación anual 2022 – 2023 versión 2**

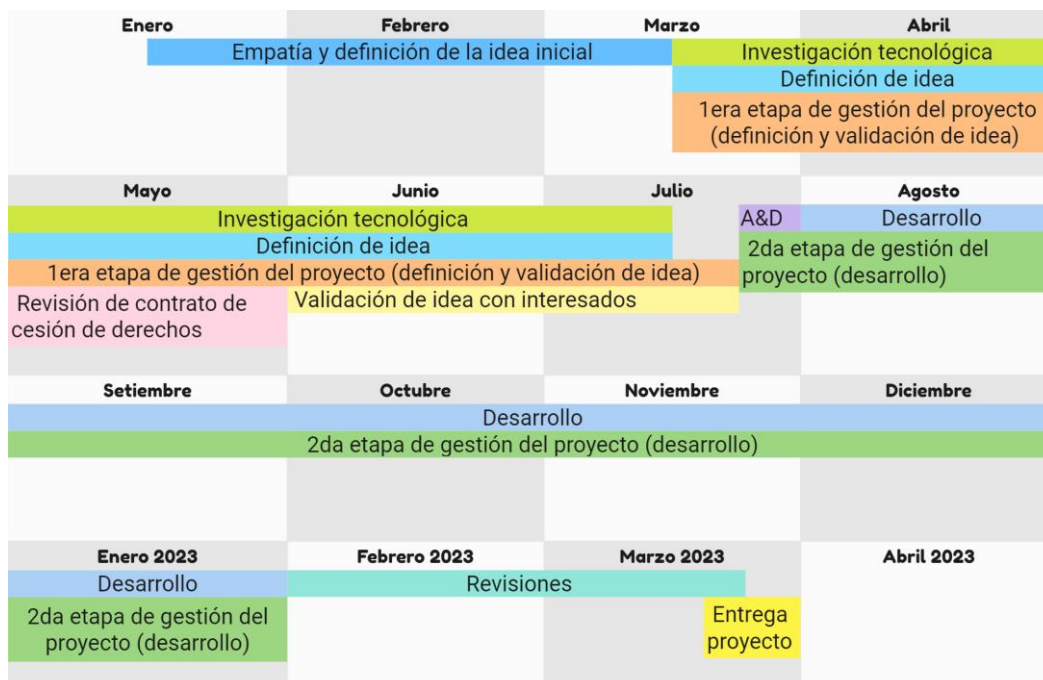


Figura 91 - Planeación anual versión 2

▪ **Planeación anual 2022 – 2023 versión 3**

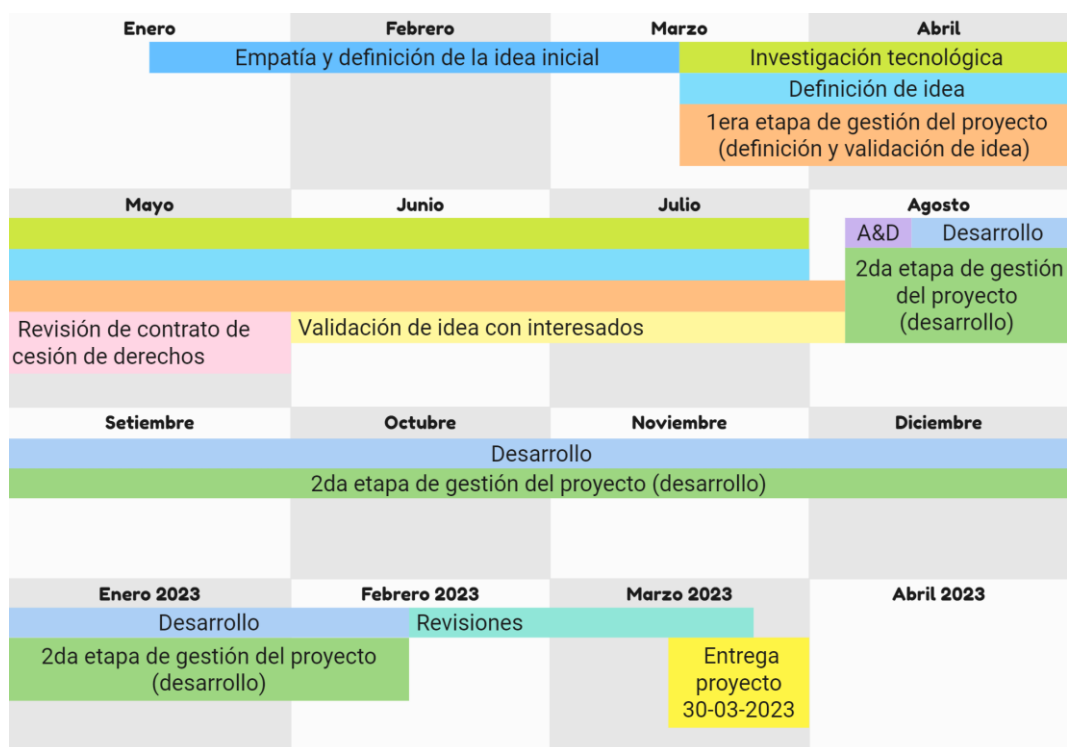


Figura 92 - Planeación anual versión 3

▪ **Planeación anual 2022 – 2023 versión 4 (final)**

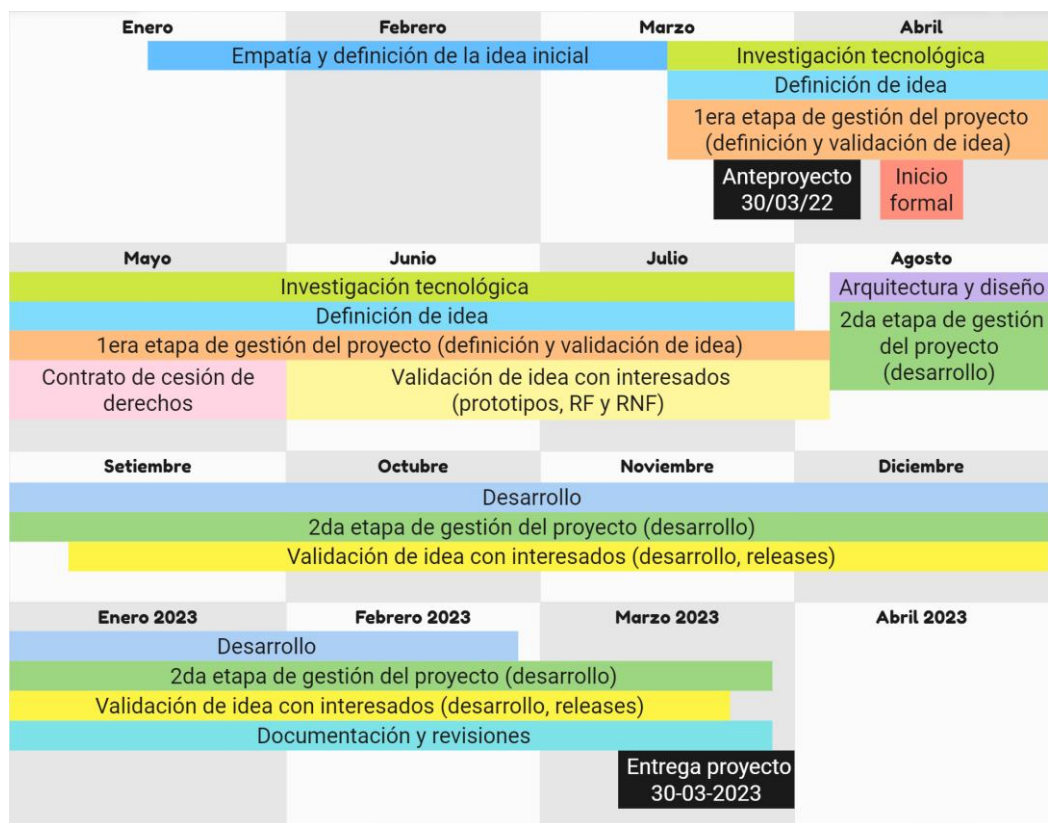


Figura 93 - Planeación anual versión 4 y final

13.15. Anexo 15: Detalle de Sprints

A continuación, detallamos los Sprints realizados a lo largo de la etapa de construcción.

- **Release 1**

SPRINT 3 - 31 Agosto al 7 de Septiembre			
User Stories			
TBMJIRA-8	Videos de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-9	[FE] Instalación de librerías básicas	2	SI
TBMJIRA-10	[BE] Instalación de librerías básicas	2	SI
TBMJIRA-11	[CD] Configuración de AWS (IAM, usuarios y roles)	8	SI
TBMJIRA-12	[CD] Configuración de serverless para deploy	5	SI
TBMJIRA-18	División en módulos de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-19	[BE] Configuración de AWS CLI	5	SI
TBMJIRA-20	[BE] Construcción de arquitectura a gran escala	8	No
Story point planeados		30	
Story point hechos		22	
Story point restantes		8	
SPRINT 4 - 7 de Septiembre al 14 de Septiembre			
User Stories			
TBMJIRA-18	División en módulos de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-21	[FE] Construcción de arquitectura a gran escala	8	SI
TBMJIRA-22	[BE] Configuración de CI y CD	5	No
TBMJIRA-23	[FE] Configuración de CI y CD	5	SI
TBMJIRA-24	[CD] Costos de CI y CD en AWS para BE provocaron que se tuvieran que hacer con GitHub Actions	5	SI
TBMJIRA-20	[BE] Construcción de arquitectura a gran escala	8	SI

Story point planeados		31	
Story point hechos		26	
Story point restantes		5	
SPRINT 5 - 14 de Septiembre al 21 de Septiembre			
User Stories			
TBMJIRA-18	División en módulos de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-22	[BE] Configuración de CI y CD	5	SI
TBMJIRA-25	[BE] Creación de endpoint para obtener información de un curso	5	SI
TBMJIRA-8	Videos de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-13	[FE] Creación de barra de menú	5	No
TBMJIRA-14	[BE] Creación de endpoint para obtener información de un video de un curso.	13	SI
TBMJIRA-15	[CD] Creación del bucket en S3	3	SI
Story point planeados		31	
Story point hechos		26	
Story point restantes		5	
SPRINT 6 - 21 de Septiembre al 28 de Septiembre			
User Stories			
TBMJIRA-8	Videos de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-13	[FE] Creación de barra de menú	5	No
TBMJIRA-16	[FE] Creación de la pantalla	8	SI
TBMJIRA-18	División en módulos de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-26	[FE] Creación de la pantalla	5	SI
TBMJIRA-27	[BE] Creación de test para el endpoint	3	SI
TBMJIRA-28	Distintos ángulos de cámara de un curso		
Tareas seleccionadas			

ID	Nombre	SP	Realizado
TBMJIRA-29	[FE] Agregación de varios reproductores de video	8	SI
Story point planeados		29	
Story point hechos		24	
Story point restantes		5	
SPRINT 7 - 28 de Septiembre al 5 de Octubre			
User Stories			
TBMJIRA-8	Videos de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-13	[FE] Creación de barra de menú	5	No
TBMJIRA-17	[BE] Creación de test para el endpoint	5	SI
TBMJIRA-30	Vista previa de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-31	[FE] Creación de la pantalla	8	SI
TBMJIRA-32	[BE] Creación de endpoint para obtener datos	5	SI
TBMJIRA-33	[BE] Creación de test para el endpoint	3	SI
Story point planeados		26	
Story point hechos		21	
Story point restantes		5	
SPRINT 8 - 5 de Octubre al 19 de Octubre			
User Stories			
TBMJIRA-8	Videos de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-13	[FE] Creación de barra de menú	5	SI
TBMJIRA-34	Sección de preguntas de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-35	[FE] Creación de la pantalla	8	SI
TBMJIRA-36	[BE] Creación de endpoint para obtener datos	8	SI
TBMJIRA-37	[BE] Creación de test para el endpoint	3	SI
TBMJIRA-38	Archivos informativos de un curso		

Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-39	[FE] Creación de la pantalla	8	SI
TBMJIRA-40	[BE] Creación de endpoint para obtener datos	5	SI
TBMJIRA-41	[BE] Creación de test para el endpoint	3	SI
Story point planeados		40	
Story point hechos		40	
Story point restantes		0	

Tabla 80 - Sprints correspondientes al Release 1

- **Release 2**

SPRINT 9 - 19 de Octubre al 2 de Noviembre			
User Stories			
TBMJIRA-42	Implementación de recorrido virtual usando video 360		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-43	[FE] Creación de componente de reproducción de video 360	34	SI
TBMJIRA-44	[CD] Configuración de MediaConvert para reproducción de videos 360	21	SI
TBMJIRA-45	[BE] Integración con servicios de video 360	21	SI
Bugs seleccionados			
ID	Nombre	SP	Realizado
TBMJIRA-46	[BE] Dependencia cíclica.	5	SI (no estaba planeado para este Sprint)
TBMJIRA-47	[FE] Video de un curso carga un video de otro curso	2	SI (no estaba planeado para este Sprint)
TBMJIRA-48	[FE] En prod se visualiza el código fuente del proyecto [PROD]	1	SI (no estaba planeado para este Sprint)
Story point planeados		76	
Story point hechos		84	
Story point restantes		0	
SPRINT 10 - 2 de Noviembre al 16 de Noviembre			
User Stories			

TBMJIRA-49 Preguntas a lo largo del recorrido virtual			
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-50	[FE] Crear el componente del modal para preguntas	13	SI
TBMJIRA-51	[FE] Conectar modal con video 360	13	SI
TBMJIRA-52	[BE] Asociar lista de preguntas a una lección de tipo recorrido virtual	8	SI
TBMJIRA-53	[FE] Conectar el frontend con el backend	5	SI
Bugs seleccionados			
ID	Nombre	SP	Realizado
TBMJIRA-54	[BE] No funciona el debug de tests unitarios	2	SI
TBMJIRA-55	[BE] Links dentro de un curso no pueden repetirse	3	SI
TBMJIRA-56	[FE][BE] Descarga de archivos pdf no funciona	8	No
TBMJIRA-57	[FE] Reproductores de videos se ven desalineados cuando se muestra más de un ángulo de cámara	5	No
Story point planeados		57	
Story point hechos		44	
Story point restantes		13	
SPRINT 11 - 16 de Noviembre al 30 de Noviembre			
User Stories			
TBMJIRA-58 Sección de explicación de funcionamiento del recorrido virtual			
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-59	[FE] Creación de componente de guía para recorridos virtuales	3	SI
Bugs seleccionados			
ID	Nombre	SP	Realizado
TBMJIRA-56	[FE][BE] Descarga de archivos pdf no funciona	8	SI
TBMJIRA-57	[FE] Reproductores de videos se ven desalineados cuando se muestra más de un ángulo de cámara	5	SI
TBMJIRA-60	[BE] MediaConvert: se crean dos jobs al convertir un video	8	SI
TBMJIRA-61	[BE] No se permite crear un curso con más de un link	5	SI
Validaciones seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-65	[VAL] Agregar distintas velocidades de reproducción	8	SI
TBMJIRA-66	[VAL] Poder intercambiar ángulos de cámara	8	SI

Story point planeados	45	
Story point hechos	45	
Story point restantes	0	

Tabla 81 - Sprints correspondientes al Release 2

- Release 3

SPRINT 12 - 30 de Noviembre al 14 de Diciembre			
User Stories			
TBMJIRA-69	Registro de un usuario		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-70	[FE] Creación de la pantalla	8	SI
TBMJIRA-71	[FE] Conexión con el backend	5	SI
TBMJIRA-72	[BE] Creación del endpoint	8	SI
TBMJIRA-73	Entrega de ejercicios entregables		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-74	[FE] Creación de la pantalla	2	SI
TBMJIRA-75	[FE] Conexión con el backend	1	SI
TBMJIRA-76	[BE] Creación del endpoint	5	SI
TBMJIRA-77	Detalle de ejercicio entregable de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-78	[FE] Creación de la pantalla	8	SI
TBMJIRA-79	[FE] Conexión con el backend	5	SI
TBMJIRA-80	[BE] Creación del endpoint	3	SI
TBMJIRA-81	[BE] Conexión con la BD	5	SI
Bugs seleccionados			
ID	Nombre	SP	Realizado
TBMJIRA-82	[BE] Validar que no se repita el "time" dentro de un mismo quiz	2	SI
TBMJIRA-83	[FE] Duración de las lecciones y duración total se muestran en segundos	1	SI (no estaba planeado para este Sprint)

TBMJIRA-84	[FE] Preguntas de recorrido virtual no se ven en pantalla completa	2	SI (no estaba planeado para este Sprint)
TBMJIRA-85	[FE] No se puede acceder a los documentos (materiales) de un curso	1	SI (no estaba planeado para este Sprint)
Story point planeados		52	
Story point hechos		56	
Story point restantes		0	
SPRINT 13 - 14 de Diciembre al 28 de Diciembre			
User Stories			
TBMJIRA-86	Visualización de consigna de ejercicio en lección		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-87	[FE] Creación de componente	2	SI
TBMJIRA-88	[CD] Acceso al link mediante CloudFront	3	SI
TBMJIRA-89	Descarga de ejercicio entregado		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-90	[FE] Creación de componente	2	SI
TBMJIRA-91	[CD] Descarga del archivo desde CloudFront	3	SI
TBMJIRA-92	Listado de ejercicios entregables		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-93	[FE] Creación de la pantalla	8	SI
TBMJIRA-94	[FE] Conexión con el backend	5	SI
TBMJIRA-95	[BE] Creación del endpoint	8	SI
Bugs seleccionados			
ID	Nombre	SP	Realizado
TBMJIRA-96	[CD] Configurar CloudFront para distintos ambientes	5	SI
TBMJIRA-97	[FE][BE] No funciona el agregar opinión en vista previa de un curso	13	SI
Validaciones seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-98	[VAL] Poder deshabilitar/habilitar quiz en recorrido virtual	2	SI
TBMJIRA-99	[VAL] Distinción de ángulos al cambiarlos de lugar	2	SI

Story point planeados	53	
Story point hechos	53	
Story point restantes	0	

Tabla 82 - Sprints correspondientes al Release 3

- Release 4

SPRINT 14 - 28 de Diciembre al 11 de Enero			
User Stories			
TBMJIRA-102	Listado de ejercicios entregados sin revisar		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-103	[FE] Creación de componente	8	SI
TBMJIRA-104	[FE] Conexión con el backend	3	SI
TBMJIRA-105	[BE] Creación del endpoint	5	SI
TBMJIRA-106	[CD] Agregar índice por identificador de docente	3	SI
TBMJIRA-107	Visualización de una entrega		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-108	[FE] Creación de componente	5	SI
TBMJIRA-109	[FE] Conexión con el backend	2	SI
TBMJIRA-110	[BE] Creación del endpoint	3	SI
Bugs seleccionados			
ID	Nombre	SP	Realizado
TBMJIRA-111	[FE] Bloquear DevTools en prod	2	SI
TBMJIRA-112	[BE] Setear dinámicamente IDs de preguntas de un quiz de una lección	5	SI
Validaciones seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-100	[VAL] Reproductor de video en segundo plano	2	SI
TBMJIRA-101	[VAL] Agregar breve info para distinguir los recorridos virtuales	5	SI
Story point planeados		43	
Story point hechos		43	
Story point restantes		0	
SPRINT 15 - 11 de Enero al 25 de Enero			

User Stories			
TBMJIRA-113	Agregar comentarios a una entrega		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-114	[FE] Creación de componente	8	SI
TBMJIRA-115	[BE] Creación del endpoint	5	SI
TBMJIRA-116	[FE] Conexión con el backend	5	SI
TBMJIRA-117	Visualización de comentarios en un ejercicio		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-118	[BE] Creación del endpoint	2	SI
TBMJIRA-119	[FE] Creación de componente	5	SI
TBMJIRA-120	[FE] Conexión con el backend	1	SI
Bugs seleccionados			
ID	Nombre	SP	Realizado
TBMJIRA-121	[CD] Corregir permisos de S3 para subida de ejercicios	5	SI
TBMJIRA-122	[FE] Evitar renderizado de los reproductores de video	5	SI
Validaciones seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-67	[VAL] Cambiar tamaño de reproductores de video	5	SI
TBMJIRA-124	[VAL] Agregar fomato y tamaño de entrega	1	SI
TBMJIRA-125	[VAL] Mensaje informativo para registro	2	SI
Story point planeados		44	
Story point hechos		44	
Story point restantes		0	

Tabla 83 - Sprints correspondientes al Release 4

- **Release 5**

SPRINT 16 - 25 de Enero al 8 de Febrero			
User Stories			
TBMJIRA-126	Búsqueda de un curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado

TBMJIRA-127	[CD] Configuración de OpenSearch para búsqueda por criterio	5	SI
TBMJIRA-128	[BE] Conexión con OpenSearch	2	SI
TBMJIRA-129	[FE] Creación de componente	2	SI
TBMJIRA-130	[BE] Creación del endpoint	2	SI
TBMJIRA-131	[FE] Conexión con el backend	2	SI
TBMJIRA-132	Selección de un curso como favorito		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-133	[BE] Creación del endpoints necesarios	5	SI
TBMJIRA-134	[FE] Creación de componente	2	SI
TBMJIRA-135	[FE] Conexión con el backend	1	SI
TBMJIRA-136	Listado de cursos favoritos		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-137	[BE] Creación del endpoint	5	SI
TBMJIRA-138	[FE] Creación de componente	2	SI
TBMJIRA-139	[FE] Conexión con el backend	1	SI
Bugs seleccionados			
ID	Nombre	SP	Realizado
TBMJIRA-140	[FE] Se tiene que recargar la página la primera vez que se ingresa a un vídeo o recorrido virtual de un curso para poder visualizarlo	3	SI
TBMJIRA-141	[FE] No se puede cambiar los ángulos de cámara desde el móvil	3	SI
TBMJIRA-142	[FE] Al recibir feedback de un ejercicio, si se ingresa desde los detalles del curso al ejercicio (estudiante), aparece como "pendiente de feedback"	1	SI
Validaciones seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-144	[VAL] Vista de docente y estudiante no está clara	1	SI
TBMJIRA-145	[VAL] Notificaciones	5	SI
Story point planeados		42	
Story point hechos		42	
Story point restantes		0	

Tabla 84 - Sprints correspondientes al Release 5

- Release 6

SPRINT 17 - 8 de Febrero al 22 de Febrero			
User Stories			
TBMJIRA-147	Creación de curso nuevo		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-148	[BE] Creación del endpoint	13	SI
TBMJIRA-149	[FE] Creación de componente	5	SI
TBMJIRA-150	[FE] Conexión con el backend	3	SI
TBMJIRA-151	Listado de cursos creados		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-152	[BE] Creación del endpoint	1	SI
TBMJIRA-153	[FE] Creación de componente	1	SI
TBMJIRA-154	[FE] Conexión con el backend	1	SI
TBMJIRA-155	Modificación de curso (admin)		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-156	[BE] Creación del endpoint	2	SI
TBMJIRA-157	[FE] Creación de componente	1	SI
TBMJIRA-158	[FE] Conexión con el backend	2	SI
TBMJIRA-159	Solicitud de creación de curso		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-160	[BE] Creación del endpoint	5	SI
TBMJIRA-161	[FE] Creación de componente	5	SI
TBMJIRA-162	[FE] Conexión con el backend	3	SI
TBMJIRA-163	Listado de solicitudes de creación de curso (admin)		
Tareas seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-164	[BE] Creación del endpoint	2	SI
TBMJIRA-165	[FE] Creación de componente	5	SI
TBMJIRA-166	[FE] Conexión con el backend	1	SI
Bugs seleccionados			

ID	Nombre	SP	Realizado
TBMJIRA-167	[FE] Mostrar los console log solo en DEV	1	SI
TBMJIRA-168	[FE] Recortar imagen del logo	1	SI
Validaciones seleccionadas			
ID	Nombre	SP	Realizado
TBMJIRA-173	[VAL] Cambiar tonos de ciertos colores	1	SI
TBMJIRA-174	[VAL] Cambiar datos mostrados en listado de cursos	1	SI
TBMJIRA-146	[VAL] Términos referentes a Ejercicios generan confusión	1	SI
TBMJIRA-68	[VAL] Agregar punteado breve del contenido del curso en "Acercas de"	1	SI
Story point planeados		56	
Story point hechos		56	
Story point restantes		0	

Tabla 85 - Sprints correspondientes al Release 6

13.16. Anexo 16: Registro detallado de horas de trabajo

Es importante destacar que se llevó un registro de horas de trabajo dedicadas desde el comienzo del proyecto, incluyendo la etapa de presentación de la idea inicial a la Universidad ORT, así como un registro desde el comienzo formal (a partir del mes de abril). A modo informativo, realizamos un total de 1626 horas desde esta última fecha.

Se optó por pasar dicho registro a una referencia, dado que ampliaría considerablemente la extensión del documento. La información puede encontrarse en la referencia [1].

Referencias bibliográficas

[1] Google Docs, "Registro de horas de trabajo", 2022 [Online]. Disponible: <https://docs.google.com/spreadsheets/d/1aMWuUGHjM98Y7j5-wKYavpqqEE-mIsTHk5B7Ce-9hHk/edit?usp=sharing>.

13.17. Anexo 17: Evolución de los riesgos

A continuación, detallamos la evolución de los riesgos mencionados en la sección [8.8.2. Estrategia](#), indicando su estado mes a mes, a partir de que fueron detectados.

A modo de comentario, se usaron las siguientes abreviaciones para simplificar las tablas: Impacto (I), Probabilidad de Ocurrencia (PO) y Magnitud (M).

- Abril 2022

ID	Riesgo	I	PO	M
3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.4	1.2
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.4	1.6

Tabla 86 - Riesgos detectados en Abril 2022

- Mayo 2022

ID	Riesgo	I	PO	M
1	Alcance superado: los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.	3	0.4	1.2
3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.4	1.2
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.4	1.6
5	Tecnologías incorrectas: las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.	3	0.2	0.6

Tabla 87 - Riesgos detectados en Mayo 2022

- Junio 2022

ID	Riesgo	I	PO	M
1	Alcance superado: los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.	3	0.4	1.2
2	Falta de integrantes: uno de los integrantes se enferma o le surgen imprevistos, por lo que no puede aportar al proyecto / no puede cumplir con sus tareas asignadas en tiempo y forma.	4	0.6	2.4

3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.4	1.2
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.4	1.6
5	Tecnologías incorrectas: las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.	3	0.2	0.6
6	Interesados en desacuerdo: los interesados no están de acuerdo con la idea propuesta, por lo que dejan de brindar apoyo.	4	0.2	0.8
7	Interesados no disponibles: los interesados son difíciles de contactar y por lo tanto, coordinar reuniones de validación con ellos.	3	0.6	1.8

Tabla 88 - Riesgos detectados en Junio 2022

- Julio 2022

ID	Riesgo	I	PO	M
1	Alcance superado: los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.	3	0.2	0.6
2	Falta de integrantes: uno de los integrantes se enferma o le surgen imprevistos, por lo que no puede aportar al proyecto / no puede cumplir con sus tareas asignadas en tiempo y forma.	4	0.6	2.4
3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.2	0.6
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.4	1.6
5	Tecnologías incorrectas: las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.	3	0.2	0.6
6	Interesados en desacuerdo: los interesados no están de acuerdo con la idea propuesta, por lo que dejan de brindar apoyo.	4	0.2	0.8
7	Interesados no disponibles: los interesados son difíciles de contactar y por lo tanto, coordinar reuniones de validación con ellos.	3	0.6	1.8

12	No conseguir obras para grabar los recorridos: no se consigue acceso a obras de construcción, de modo que no podemos obtener grabaciones realistas.	3	0.2	0.6
13	No conseguir cámara 360: no se consigue una cámara capaz de grabar videos 360, de modo que no podemos conseguir tener recorridos virtuales en obras realistas.	3	0.2	0.6

Tabla 89 - Riesgos detectados en Julio 2022

- Agosto 2022

ID	Riesgo	I	PO	M
1	Alcance superado: los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.	3	0.2	0.6
2	Falta de integrantes: uno de los integrantes se enferma o le surgen imprevistos, por lo que no puede aportar al proyecto / no puede cumplir con sus tareas asignadas en tiempo y forma.	4	0.6	2.4
3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.2	0.6
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.4	1.6
5	Tecnologías incorrectas: las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.	3	0.2	0.6
6	Interesados en desacuerdo: los interesados no están de acuerdo con la idea propuesta, por lo que dejan de brindar apoyo.	4	0.2	0.8
7	Interesados no disponibles: los interesados son difíciles de contactar y por lo tanto, coordinar reuniones de validación con ellos.	3	0.6	1.8
12	No conseguir obras para grabar los recorridos: no se consigue acceso a obras de construcción, de modo que no podemos obtener grabaciones realistas.	3	0.2	0.6
13	No conseguir cámara 360: no se consigue una cámara capaz de grabar videos 360, de modo que no podemos conseguir tener recorridos virtuales en obras realistas.	3	0.2	0.6

Tabla 90 - Riesgos detectados en Agosto 2022

- Setiembre 2022

ID	Riesgo	I	PO	M
1	Alcance superado: los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.	3	0.2	0.6
2	Falta de integrantes: uno de los integrantes se enferma o le surgen imprevistos, por lo que no puede aportar al proyecto / no puede cumplir con sus tareas asignadas en tiempo y forma.	4	0.6	2.4
3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.2	0.6
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.2	0.8
5	Tecnologías incorrectas: las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.	3	0.2	0.6
6	Interesados en desacuerdo: los interesados no están de acuerdo con la idea propuesta, por lo que dejan de brindar apoyo.	4	0.2	0.8
7	Interesados no disponibles: los interesados son difíciles de contactar y por lo tanto, coordinar reuniones de validación con ellos.	3	0.6	1.8
8	Arquitectura cambiante: debido a la naturaleza de nuestro proyecto, los cambios son inevitables, por lo que la arquitectura puede verse alterada.	3	0.4	1.2
9	Objetivos no alcanzados en sprint: los objetivos del sprint no son alcanzados debido a imprevistos que impidan o retrasen lo planeado, por lo que las tareas estimadas se atrasan.	3	0.4	1.2
10	Problemas con servicios Cloud: alguno de los servicios que utilizemos remotamente falla, prohibiéndonos utilizarlo o configurarlo remotamente.	4	0.6	2.4
11	Mala estimación de la duración de tareas: mala estimación de la duración a la hora de realizar los cambios, haciendo que demoren más de lo previsto.	3	0.4	1.2
12	No conseguir obras para grabar los recorridos: no se consigue acceso a obras de construcción, de modo que no podemos obtener grabaciones realistas.	3	0.2	0.6
13	No conseguir cámara 360: no se consigue una cámara capaz de grabar videos 360, de modo que no podemos conseguir tener recorridos virtuales en obras realistas.	3	0.2	0.6

Tabla 91 - Riesgos detectados en Setiembre 2022

▪ Octubre 2022

ID	Riesgo	I	PO	M
1	Alcance superado: los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.	3	0.2	0.6
2	Falta de integrantes: uno de los integrantes se enferma o le surgen imprevistos, por lo que no puede aportar al proyecto / no puede cumplir con sus tareas asignadas en tiempo y forma.	4	0.4	1.6
3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.2	0.6
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.2	0.8
5	Tecnologías incorrectas: las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.	3	0.2	0.6
6	Interesados en desacuerdo: los interesados no están de acuerdo con la idea propuesta, por lo que dejan de brindar apoyo.	4	0.2	0.8
7	Interesados no disponibles: los interesados son difíciles de contactar y por lo tanto, coordinar reuniones de validación con ellos.	3	0.6	1.8
8	Arquitectura cambiante: debido a la naturaleza de nuestro proyecto, los cambios son inevitables, por lo que la arquitectura puede verse alterada.	3	0.2	0.6
9	Objetivos no alcanzados en sprint: los objetivos del sprint no son alcanzados debido a imprevistos que impidan o retrasen lo planeado, por lo que las tareas estimadas se atrasan.	3	0.4	1.2
10	Problemas con servicios Cloud: alguno de los servicios que utilizemos remotamente falla, prohibiéndonos utilizarlo o configurarlo remotamente.	4	0.4	1.6
11	Mala estimación de la duración de tareas: mala estimación de la duración a la hora de realizar los cambios, haciendo que demoren más de lo previsto.	3	0.4	1.2
12	No conseguir obras para grabar los recorridos: no se consigue acceso a obras de construcción, de modo que no podemos obtener grabaciones realistas.	3	0.2	0.6

13	No conseguir cámara 360: no se consigue una cámara capaz de grabar videos 360, de modo que no podemos conseguir tener recorridos virtuales en obras realistas.	3	0.2	0.6
14	Falta de fondos en AWS: debido a un alto uso de los servicios de AWS, se consume todo el dinero de los fondos académicos que nos otorgaron.	4	0.2	0.8

Tabla 92 - Riesgos detectados en Octubre 2022

- Noviembre 2022

ID	Riesgo	I	PO	M
1	Alcance superado: los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.	3	0.2	0.6
2	Falta de integrantes: uno de los integrantes se enferma o le surgen imprevistos, por lo que no puede aportar al proyecto / no puede cumplir con sus tareas asignadas en tiempo y forma.	4	0.4	1.6
3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.2	0.6
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.2	0.8
5	Tecnologías incorrectas: las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.	3	0.2	0.6
6	Interesados en desacuerdo: los interesados no están de acuerdo con la idea propuesta, por lo que dejan de brindar apoyo.	4	0.2	0.8
7	Interesados no disponibles: los interesados son difíciles de contactar y por lo tanto, coordinar reuniones de validación con ellos.	3	0.6	1.8
8	Arquitectura cambiante: debido a la naturaleza de nuestro proyecto, los cambios son inevitables, por lo que la arquitectura puede verse alterada.	3	0.2	0.6
9	Objetivos no alcanzados en sprint: los objetivos del sprint no son alcanzados debido a imprevistos que impidan o retrasen lo planeado, por lo que las tareas estimadas se atrasan.	3	0.2	0.6
10	Problemas con servicios Cloud: alguno de los servicios que utilicemos remotamente falla, prohibiéndonos utilizarlo o configurarlo remotamente.	4	0.4	1.6

11	Mala estimación de la duración de tareas: mala estimación de la duración a la hora de realizar los cambios, haciendo que demoren más de lo previsto.	3	0.2	0.6
12	No conseguir obras para grabar los recorridos: no se consigue acceso a obras de construcción, de modo que no podemos obtener grabaciones realistas.	3	0.2	0.6
13	No conseguir cámara 360: no se consigue una cámara capaz de grabar videos 360, de modo que no podemos conseguir tener recorridos virtuales en obras realistas.	3	0.2	0.6
14	Falta de fondos en AWS: debido a un alto uso de los servicios de AWS, se consume todo el dinero de los fondos académicos que nos otorgaron.	4	0.2	0.8

Tabla 93 - Riesgos detectados en Noviembre 2022

- Diciembre 2022

ID	Riesgo	I	PO	M
1	Alcance superado: los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.	3	0.2	0.6
2	Falta de integrantes: uno de los integrantes se enferma o le surgen imprevistos, por lo que no puede aportar al proyecto / no puede cumplir con sus tareas asignadas en tiempo y forma.	4	0.4	1.6
3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.2	0.6
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.2	0.8
5	Tecnologías incorrectas: las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.	3	0.2	0.6
6	Interesados en desacuerdo: los interesados no están de acuerdo con la idea propuesta, por lo que dejan de brindar apoyo.	4	0.2	0.8
7	Interesados no disponibles: los interesados son difíciles de contactar y por lo tanto, coordinar reuniones de validación con ellos.	3	0.8	2.4
8	Arquitectura cambiante: debido a la naturaleza de nuestro proyecto, los cambios son inevitables, por lo que la arquitectura puede verse alterada.	3	0.2	0.6

9	Objetivos no alcanzados en sprint: los objetivos del sprint no son alcanzados debido a imprevistos que impidan o retrasen lo planeado, por lo que las tareas estimadas se atrasan.	3	0.2	0.6
10	Problemas con servicios Cloud: alguno de los servicios que utilizemos remotamente falla, prohibiéndonos utilizarlo o configurarlo remotamente.	4	0.2	0.8
11	Mala estimación de la duración de tareas: mala estimación de la duración a la hora de realizar los cambios, haciendo que demoren más de lo previsto.	3	0.2	0.6
12	No conseguir obras para grabar los recorridos: no se consigue acceso a obras de construcción, de modo que no podemos obtener grabaciones realistas.	2	0.6	1.2
13	No conseguir cámara 360: no se consigue una cámara capaz de grabar videos 360, de modo que no podemos conseguir tener recorridos virtuales en obras realistas.	2	0.2	0.4
14	Falta de fondos en AWS: debido a un alto uso de los servicios de AWS, se consume todo el dinero de los fondos académicos que nos otorgaron.	4	0.2	0.8

Tabla 94 - Riesgos detectados en Diciembre 2022

- Enero 2023

ID	Riesgo	I	PO	M
1	Alcance superado: los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.	3	0.2	0.6
2	Falta de integrantes: uno de los integrantes se enferma o le surgen imprevistos, por lo que no puede aportar al proyecto / no puede cumplir con sus tareas asignadas en tiempo y forma.	4	0.2	0.8
3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.2	0.6
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.2	0.8
5	Tecnologías incorrectas: las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.	3	0.2	0.6
6	Interesados en desacuerdo: los interesados no están de acuerdo con la idea propuesta, por lo que dejan de brindar apoyo.	4	0.2	0.8

7	Interesados no disponibles: los interesados son difíciles de contactar y por lo tanto, coordinar reuniones de validación con ellos.	3	0.8	2.4
8	Arquitectura cambiante: debido a la naturaleza de nuestro proyecto, los cambios son inevitables, por lo que la arquitectura puede verse alterada.	3	0.2	0.6
9	Objetivos no alcanzados en sprint: los objetivos del sprint no son alcanzados debido a imprevistos que impidan o retrasen lo planeado, por lo que las tareas estimadas se atrasan.	3	0.2	0.6
10	Problemas con servicios Cloud: alguno de los servicios que utilicemos remotamente falla, prohibiéndonos utilizarlo o configurarlo remotamente.	4	0.2	0.8
11	Mala estimación de la duración de tareas: mala estimación de la duración a la hora de realizar los cambios, haciendo que demoren más de lo previsto.	3	0.2	0.6
12	No conseguir obras para grabar los recorridos: no se consigue acceso a obras de construcción, de modo que no podemos obtener grabaciones realistas.	2	0.6	1.2
13	No conseguir cámara 360: no se consigue una cámara capaz de grabar videos 360, de modo que no podemos conseguir tener recorridos virtuales en obras realistas.	2	0.2	0.4
14	Falta de fondos en AWS: debido a un alto uso de los servicios de AWS, se consume todo el dinero de los fondos académicos que nos otorgaron.	4	0.2	0.8

Tabla 95 - Riesgos detectados en Enero 2023

- Febrero 2023

ID	Riesgo	I	PO	M
1	Alcance superado: los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.	3	0.2	0.6
2	Falta de integrantes: uno de los integrantes se enferma o le surgen imprevistos, por lo que no puede aportar al proyecto / no puede cumplir con sus tareas asignadas en tiempo y forma.	5	0.2	1.0
3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.2	0.6
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.2	0.8

5	Tecnologías incorrectas: las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.	3	0.2	0.6
6	Interesados en desacuerdo: los interesados no están de acuerdo con la idea propuesta, por lo que dejan de brindar apoyo.	4	0.2	0.8
7	Interesados no disponibles: los interesados son difíciles de contactar y por lo tanto, coordinar reuniones de validación con ellos.	3	0.6	1.8
8	Arquitectura cambiante: debido a la naturaleza de nuestro proyecto, los cambios son inevitables, por lo que la arquitectura puede verse alterada.	3	0.2	0.6
9	Objetivos no alcanzados en sprint: los objetivos del sprint no son alcanzados debido a imprevistos que impidan o retrasen lo planeado, por lo que las tareas estimadas se atrasan.	3	0.2	0.6
10	Problemas con servicios Cloud: alguno de los servicios que utilizemos remotamente falla, prohibiéndonos utilizarlo o configurarlo remotamente.	4	0.2	0.8
11	Mala estimación de la duración de tareas: mala estimación de la duración a la hora de realizar los cambios, haciendo que demoren más de lo previsto.	3	0.2	0.6
12	No conseguir obras para grabar los recorridos: no se consigue acceso a obras de construcción, de modo que no podemos obtener grabaciones realistas.	2	0.2	0.4
13	No conseguir cámara 360: no se consigue una cámara capaz de grabar videos 360, de modo que no podemos conseguir tener recorridos virtuales en obras realistas.	2	0.2	0.4
14	Falta de fondos en AWS: debido a un alto uso de los servicios de AWS, se consume todo el dinero de los fondos académicos que nos otorgaron.	4	0.2	0.8

Tabla 96 - Riesgos detectados en Febrero 2023

- Marzo 2023

ID	Riesgo	I	PO	M
1	Alcance superado: los RF y/o RNF definidos superan el alcance del proyecto y por lo tanto, no se llegan a cumplir con todos en el límite de tiempo establecido.	3	0.2	0.6
2	Falta de integrantes: uno de los integrantes se enferma o le surgen imprevistos, por lo que no puede aportar al proyecto / no puede cumplir con sus tareas asignadas en tiempo y forma.	5	0.2	1.0

3	Disputas entre integrantes: se generan problemas de comunicación entre los integrantes del equipo, resultando en malentendidos y retrasos.	3	0.2	0.6
4	Atrasos por desconocimiento del dominio: el poco conocimiento del equipo acerca del dominio provoca atrasos en el proyecto.	4	0.2	0.8
5	Tecnologías incorrectas: las tecnologías seleccionadas no resultan ser las adecuadas para el proyecto, provocando retrasos y resultados malos.	3	0.2	0.6
6	Interesados en desacuerdo: los interesados no están de acuerdo con la idea propuesta, por lo que dejan de brindar apoyo.	4	0.2	0.8
7	Interesados no disponibles: los interesados son difíciles de contactar y por lo tanto, coordinar reuniones de validación con ellos.	3	0.6	1.8
8	Arquitectura cambiante: debido a la naturaleza de nuestro proyecto, los cambios son inevitables, por lo que la arquitectura puede verse alterada.	2	0.2	0.4
10	Problemas con servicios Cloud: alguno de los servicios que utilizamos remotamente falla, prohibiéndonos utilizarlo o configurarlo remotamente.	4	0.2	0.8
12	No conseguir obras para grabar los recorridos: no se consigue acceso a obras de construcción, de modo que no podemos obtener grabaciones realistas.	2	0.2	0.4
13	No conseguir cámara 360: no se consigue una cámara capaz de grabar videos 360, de modo que no podemos conseguir tener recorridos virtuales en obras realistas.	2	0.2	0.4
14	Falta de fondos en AWS: debido a un alto uso de los servicios de AWS, se consume todo el dinero de los fondos académicos que nos otorgaron.	4	0.2	0.8

Tabla 97 - Riesgos detectados en Marzo 2023

13.18. Anexo 18: Template de Pull Request

A continuación, se muestra el Template a seguir para las Pull Request creadas en los repositorios de Backend y Frontend (Figura 94).

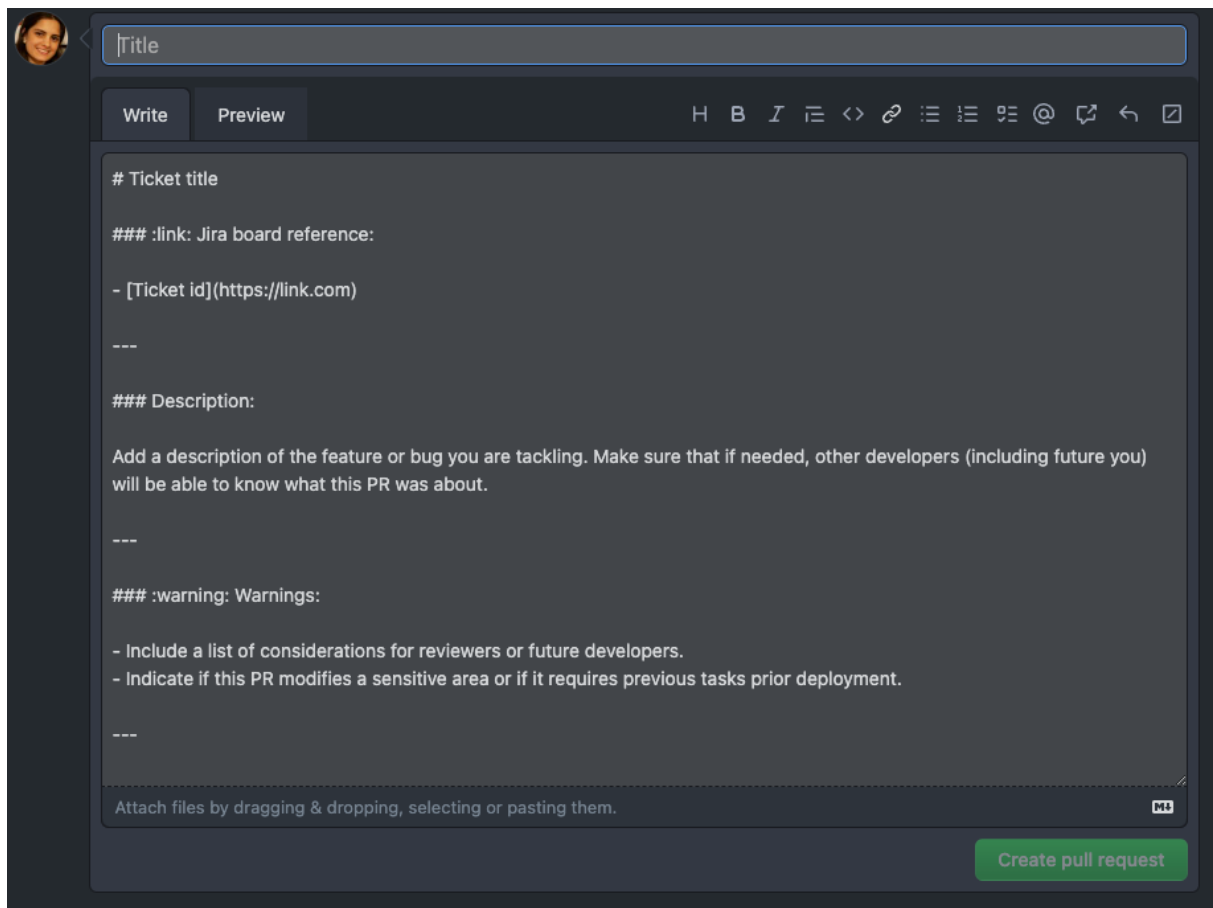


Figura 94 - Template de Pull Request

A gran escala podemos ver que contiene título y ticket de Jira (pudiendo ser uno o más tickets relacionados), una descripción del trabajo realizado y advertencias a tener en cuenta, sobre consideraciones futuras, si requiere tareas previas o si puede tocar alguna parte sensible del proyecto.

13.19. Anexo 19: Librerías externas

A continuación, se definen las librerías externas utilizadas más importantes.

- aws-amplify

Librería provista por AWS que nos permite acceder a nuestros servicios desde nuestra aplicación web.

- aws-jwt-verify

Librería provista por AWS que nos permite validar y obtener el payload de un JWT generado por AWS Cognito.

- aws-sdk

Librería provista por AWS que nos permite conectarnos con otros servicios de AWS desde una función, en nuestro caso.

- eslint

Librería que nos permite bloquear la ejecución o mostrar warnings a la hora de correr un servicio o nuestra aplicación, con los errores de tipado o variables sin usar, dependiendo la configuración establecida en su archivo de configuración.

- express

Librería que nos permite generar una API, configurando rutas, CORS, leyendo las solicitudes y armando respuestas HTTP.

- husky

Librería que nos permite bloquear ciertas acciones en caso de que no se cumplan determinadas precondiciones de forma local. En nuestro caso, nos bloquea los commits y push a una rama en caso de que se contenga errores de ESLint o nuestra función/aplicación web no funcione.

- material-ui

Como ya se detalló anteriormente, esta librería nos permite seguir los lineamientos del sistema de diseño Material Design.

- mocha-tests

Librería utilizada para la generación de test unitarios a realizar en las funciones.

- prettier

Librería utilizada para establecer el formato de los archivos, configurando el espaciado y indentado por defecto a seguir en su archivo de configuración entre otras propiedades.

- react-i18next

Librería que nos permite leer y mantener agrupadas las constantes de texto mostradas a los usuarios, pudiendo detectar la ubicación del sistema y mostrar un idioma u otro en base a esta.

- react-redux

Librería que nos permite generar el contexto con Redux como se menciona en la sección [7.3.2. Manejo de datos en el Frontend](#).

- uuid

Librería encargada de generar identificadores únicos, en nuestro caso, utilizada para la creación de cursos.

- video.js

Librería que nos permite la reproducción de video a demanda.

13.20. Anexo 20: Feedback de Releases

Se optó por pasar los comentarios detallados, obtenidos de las diferentes instancias de validación mencionadas en la sección [10.3.3. Releases](#), a una referencia, puesto que ampliaría considerablemente la extensión del documento. La información puede encontrarse en la referencia [1].

Referencias bibliográficas

[1] Google Docs, “Validaciones Releases”, 2022 [Online]. Disponible:

https://docs.google.com/spreadsheets/d/1GiV_KiEnFP0fmDx_kkrWpzJfuxnyfW3amrfUlGo5gok/edit?usp=sharing.

13.21. Anexo 21: Tareas de las validaciones

A continuación, detallamos todas las tareas creadas a partir de las diferentes instancias de validación de los Releases, como se menciona en la sección [10.3.3. Releases](#).

A modo de comentario, en la siguiente tabla se refiere a la prioridad de la tarea con la letra “P” y a los Story Points como “SP”.

ID	Nombre	P	Origen	Sprint en que se realizó	SP
TBMJIRA-62	[VAL] Agregar descripción breve a cada lección	Baja	Release 1	Pendiente	-
TBMJIRA-63	[VAL] Detalle de herramientas, máquinas y/o marcas utilizadas en VR	Baja	Release 1	Pendiente	-
TBMJIRA-64	[VAL] Coordinar videos cuando hay distintos ángulos de cámara	Media	Release 1	Pendiente	-
TBMJIRA-65	[VAL] Agregar distintas velocidades de reproducción	Alta	Release 1	Sprint 11	8
TBMJIRA-66	[VAL] Poder intercambiar ángulos de cámara	Alta	Release 1	Sprint 11	8
TBMJIRA-67	[VAL] Cambiar tamaño de reproductores de video	Alta	Release 1	Sprint 15	5
TBMJIRA-68	[VAL] Agregar punteado breve del contenido del curso en "Acerca de"	Media	Release 1	Sprint 17	1
TBMJIRA-98	[VAL] Poder deshabilitar/habilitar quiz en recorrido virtual	Alta	Release 2	Sprint 13	2
TBMJIRA-99	[VAL] Distinción de ángulos al cambiarlos de lugar	Alta	Release 2	Sprint 13	2
TBMJIRA-100	[VAL] Reproductor de video en segundo plano	Media	Release 2	Sprint 14	2
TBMJIRA-101	[VAL] Agregar breve info para distinguir los recorridos virtuales	Media	Release 2	Sprint 14	5
TBMJIRA-123	[VAL] Registro y login a través de terceros	Media	Release 3	Pendiente	-
TBMJIRA-124	[VAL] Agregar fomato y tamaño de entrega	Media	Release 3	Sprint 15	1
TBMJIRA-125	[VAL] Mensaje informativo para registro	Alta	Release 3	Sprint 15	2
TBMJIRA-143	[VAL] Agregar drag&drop para subir varios archivos de una vez	Media	Release 4	Pendiente	-
TBMJIRA-144	[VAL] Vista de docente y estudiante no está clara	Media	Release 4	Sprint 16	1
TBMJIRA-145	[VAL] Notificaciones	Media	Release 4	Sprint 16	5
TBMJIRA-146	[VAL] Términos referentes a Ejercicios generan confusión	Media	Release 4	Sprint 17	1

ID	Nombre	P	Origen	Sprint en que se realizó	SP
TBMJIRA-169	[VAL] Ver quiz completo con soluciones	Media	Release 5	Pendiente	-
TBMJIRA-170	[VAL] Re-entrega de ejercicio de una lección	Media	Release 5	Pendiente	-
TBMJIRA-171	[VAL] Filtrar ejercicios por curso	Media	Release 5	Pendiente	-
TBMJIRA-172	[VAL] Agregar favoritos desde listado de cursos	Media	Release 5	Pendiente	-
TBMJIRA-173	[VAL] Cambiar tonos de ciertos colores	Media	Release 5	Sprint 17	1
TBMJIRA-174	[VAL] Cambiar datos mostrados en listado de cursos	Media	Release 5	Sprint 17	1
TBMJIRA-175	[VAL] Agregar más fomatos de archivo para solicitud de creacion de curso	Media	Release 6	Pendiente	-
TBMJIRA-176	[VAL] Edición de un curso por docente y publicación por admin	Media	Release 6	Pendiente	-
TBMJIRA-177	[VAL] Poner límite de caracteres para solicitud de creación de curso	Media	Release 6	Pendiente	-

Tabla 98 - Tareas surgidas de las validaciones

13.22. Anexo 22: Cumplimiento de las Heurísticas de Nielsen (RNF5)

A continuación, proveemos evidencia del cumplimiento de las heurísticas de Nielsen, distinguiendo cada una de ellas.

- **Visibilidad del estado del sistema**

El sistema debe mantener informado al usuario sobre lo que está sucediendo en todo momento. Para ello, por ejemplo, cuando se registra un usuario y se envía la confirmación o cuando se solicita una nueva contraseña porque se perdió la anterior, aparece un pop-up que indica el estado de éxito. A su vez, en la entrega de ejercicios o cuando se publica el curso se pone siempre visible el estado y se marca el nuevo estado al que paso el sistema. También, se indica al usuario cuando la página está cargando con un loader. A continuación mostraremos evidencia del caso en que se solicita cambiar de contraseña.



Figura 95 - H1

- **Relación entre el sistema y el mundo real**

La plataforma web debe seguir las convenciones del mundo real, de modo que los usuarios sepan intuitivamente cómo usarla. Para ello, por ejemplo, nosotros ofrecemos un buscador que funciona similar a cualquier motor de búsqueda, donde si se presiona enter o en la lupa se busca por el criterio establecido. También, los botones de subida de ejercicios o archivos en la creación de un curso tienen el mismo funcionamiento que en otros sistemas, apareciendo el pop-up para seleccionar estos. A continuación, evidenciaremos el caso del buscador de cursos.

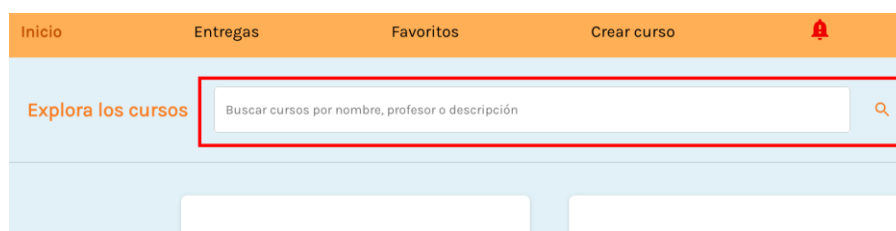


Figura 96 - H2

- **Control y libertad del usuario**

Los usuarios deben sentir que tienen control sobre el sistema en todo momento. Para ello, por ejemplo, nosotros permitimos cambiar el archivo seleccionado antes del envío formal de un ejercicio. También, al publicar un curso, este puede ser modificado. A

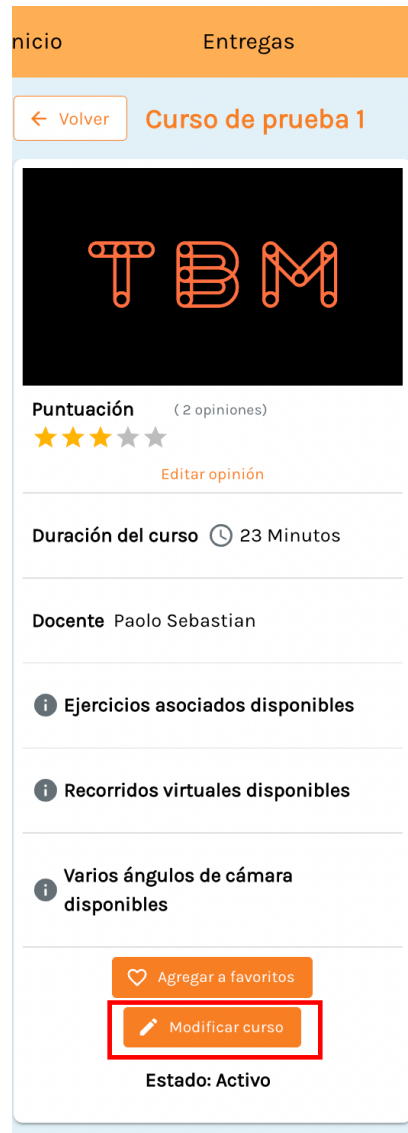


Figura 97 - H3

continuación, evidenciamos el caso de modificación de un curso.

- **Consistencia y estándares**

La plataforma web debe seguir estándares y convenciones comunes en cuanto a diseño y usabilidad. Para ello, por ejemplo, nosotros ubicamos el botón de cerrar sesión en la parte superior derecha de la pantalla. Asimismo, ubicamos el contacto en

el footer de nuestra plataforma web. A continuación, evidenciaremos el botón de cerrar sesión.

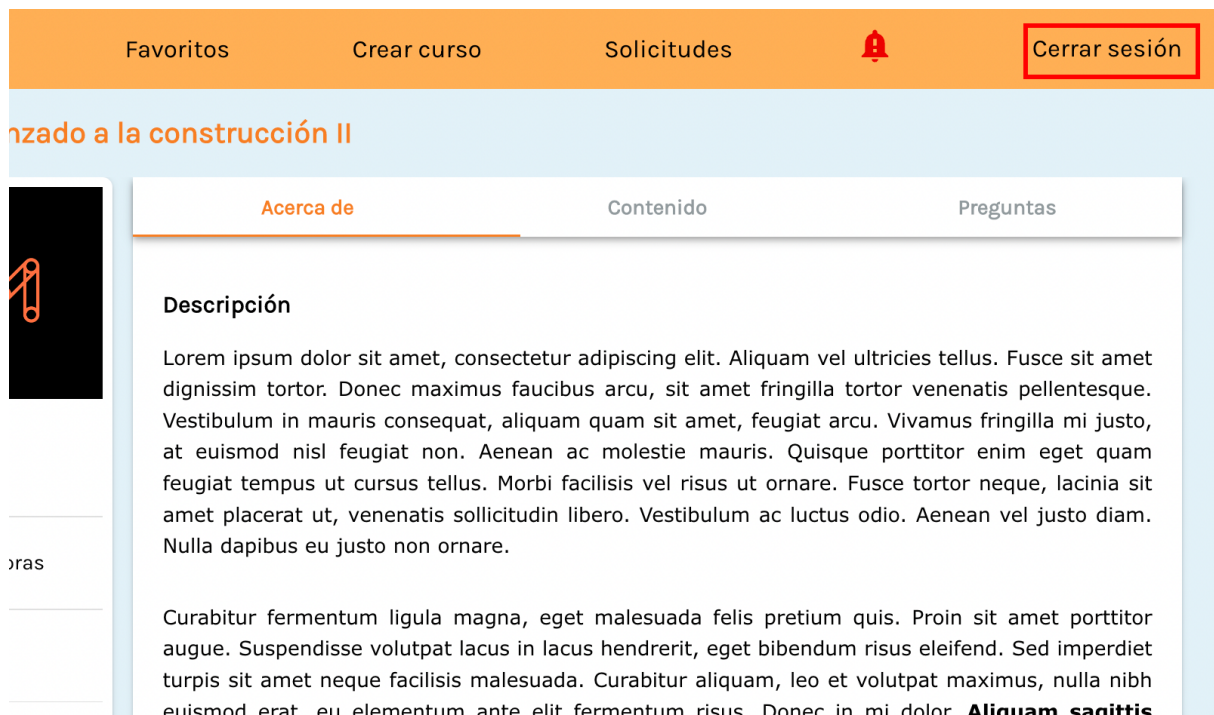


Figura 98 - H4

▪ Prevención de errores

La plataforma web debe prevenir errores en lugar de simplemente responder a ellos. Para ello, nosotros ponemos comentarios que indican como deben completarse los formularios. El mejor caso es cuando se crea la contraseña para registrarse, donde se lista todos los detalles que tiene que tener esta. A continuación, evidenciaremos el caso de completar la contraseña para el registro.



Figura 99 - H5

- **Reconocer antes que recordar**

Busca reducir la carga cognitiva de los usuarios al hacer que la información relevante esté disponible en el contexto y lugar apropiados, en lugar de obligarles a recordarla. Para ello, nosotros ponemos iconos en varios botones para que el usuario los asocie, a su vez utilizamos colores en la parte de creación de un curso para asociar las acciones. A continuación, evidenciamos el cumplimiento de esta heurística con el uso de iconos y colores en los botones en los botones.



Figura 100 - H6

- **Flexibilidad y eficiencia de uso**

El objetivo de esta heurística es el sistema para que tanto los usuarios novatos como los expertos puedan usarlo eficazmente. Para ello, nuestra plataforma es compatible por la navegación por medio del tabulador y se puede hacer enter para seleccionar un botón en lugar de hacer click. A continuación, evidenciamos el cumplimiento de esta heurística con el uso de tabulador para llegar a un botón.

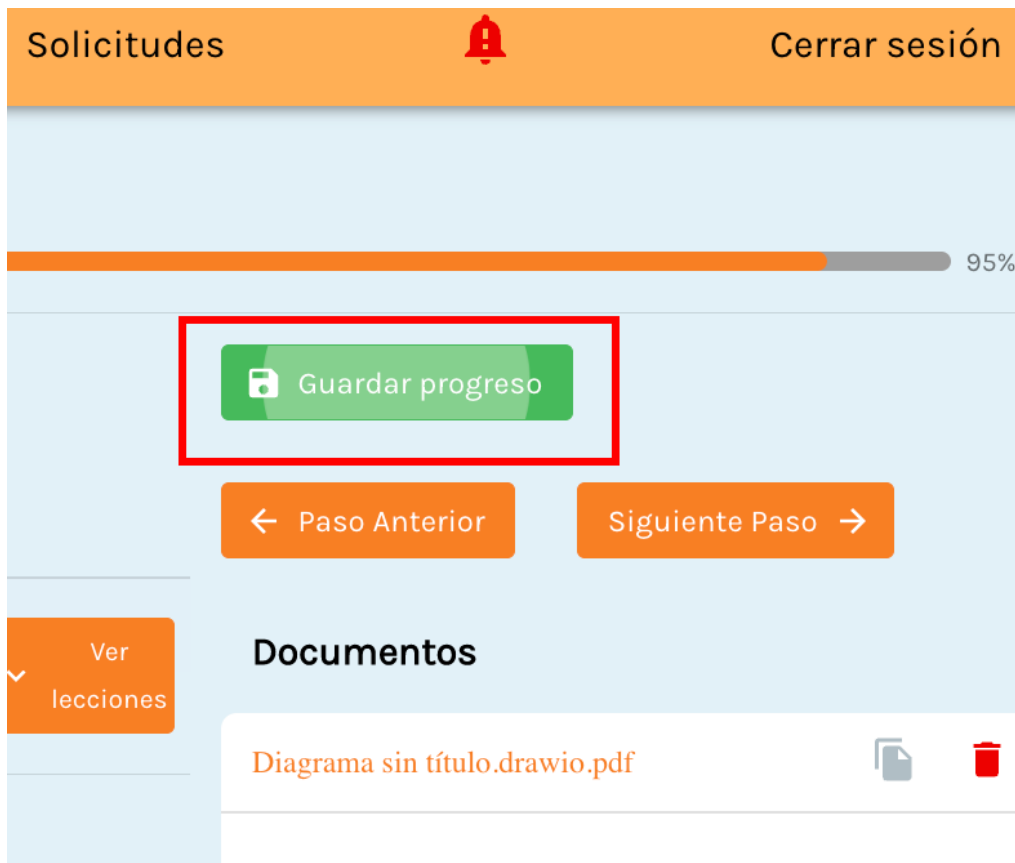


Figura 101 - H7

- **Diseño estético y minimalista**

Esta heurística busca que diseño visual y la experiencia de usuario sean atractivos y agradables para estos. Para esto, nosotros además de utilizar un Sistema de diseño ver sección [7.3.1. Sistema de diseño](#), utilizamos la fuente recomendada por este, la cual se llama Roboto. A continuación, mostramos evidencia del uso de una tipografía y Sistema de diseño alineado con esta heurística.

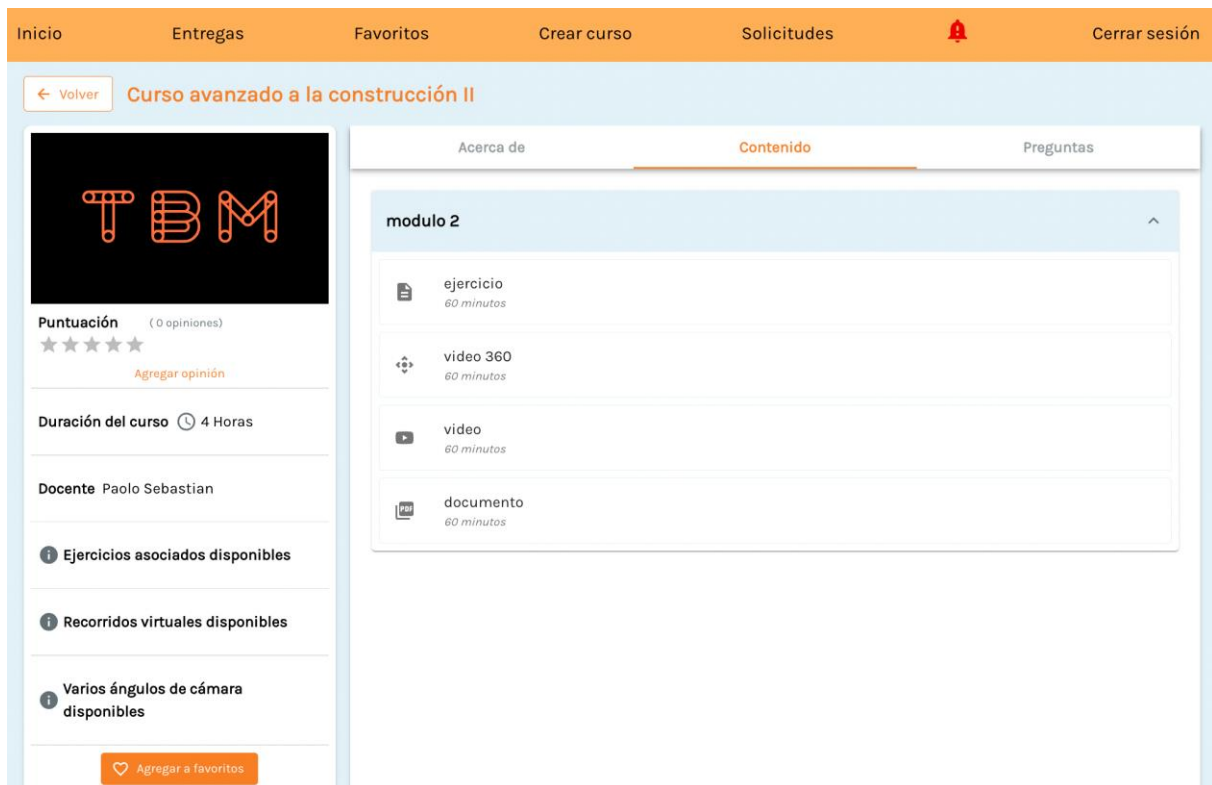


Figura 102 - H8

- **Ayudar a los usuarios a reconocer, diagnosticar y corregir los errores**

El sistema debe ayudar a los usuarios a comprender los errores y proporcionar una manera de recuperar el trabajo perdido. Para ello, la funcionalidad principal que se alinea con esta heurística es la de guardar el proceso de creación de curso para luego poder retomarlo. En las heurísticas anteriores presentamos evidencia del guardado del progreso de creación de un curso.

- **Ayuda y documentación**

Esta heurística se refiere a la importancia de tener documentación que permita entender cómo funciona el sistema. El mejor ejemplo de esto es la ayuda brindada para poder entender el funcionamiento de los videos 360. A continuación, presentamos evidencia del cumplimiento de esta heurística.

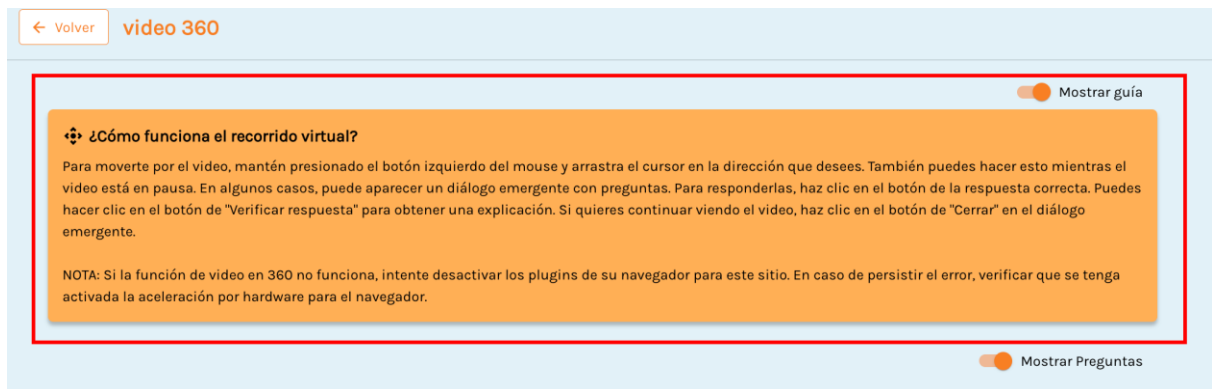


Figura 103 - H10

13.23. Anexo 23: Cumplimiento de responsiveness (RNF6)

A continuación, se brindará evidencia del cumplimiento del RNF6, en particular, mostrando capturas de la vista en detalle de un curso y la visualización de vídeos con distintos ángulos de cámara. Para ello, se tuvo como base los 400px mínimos establecidos en el criterio de aceptación y 600px con el objetivo de demostrar que se mantiene el responsive de la plataforma a medida que aumenta su ancho.

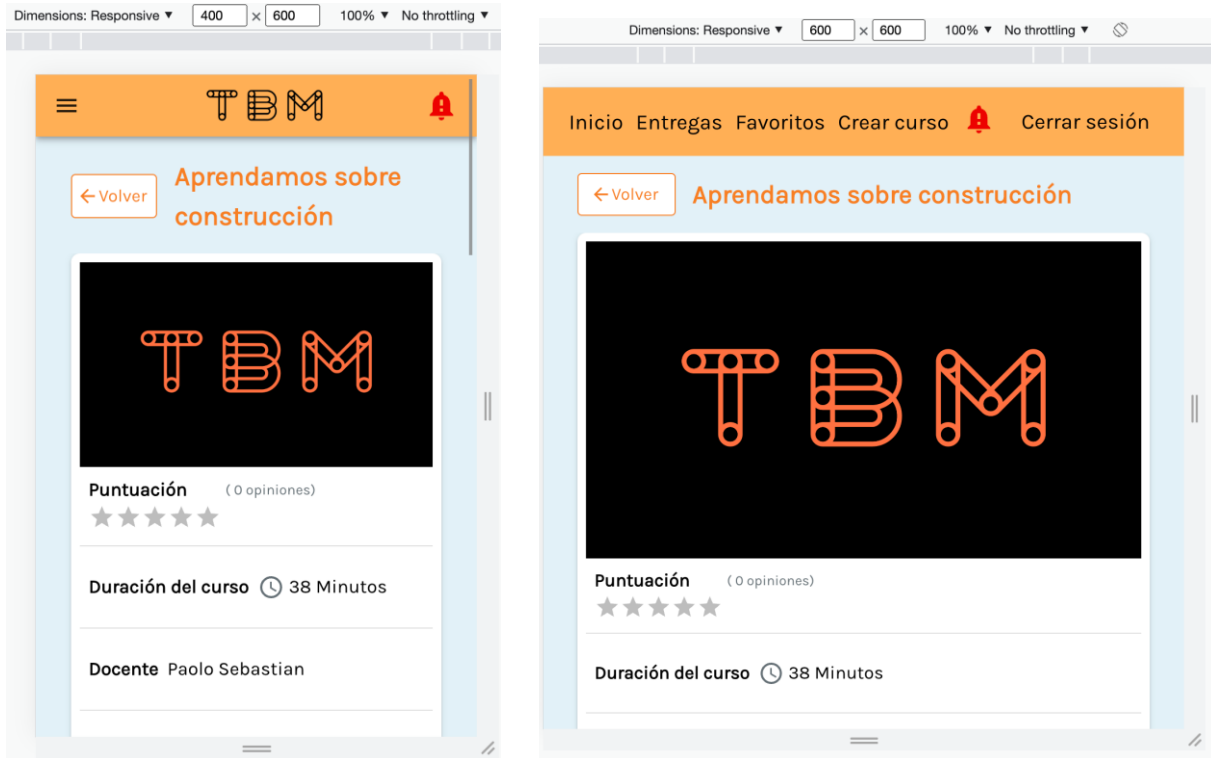


Figura 104 - Vista de detalle de un curso en 400px (izquierda) y 600px (derecha)

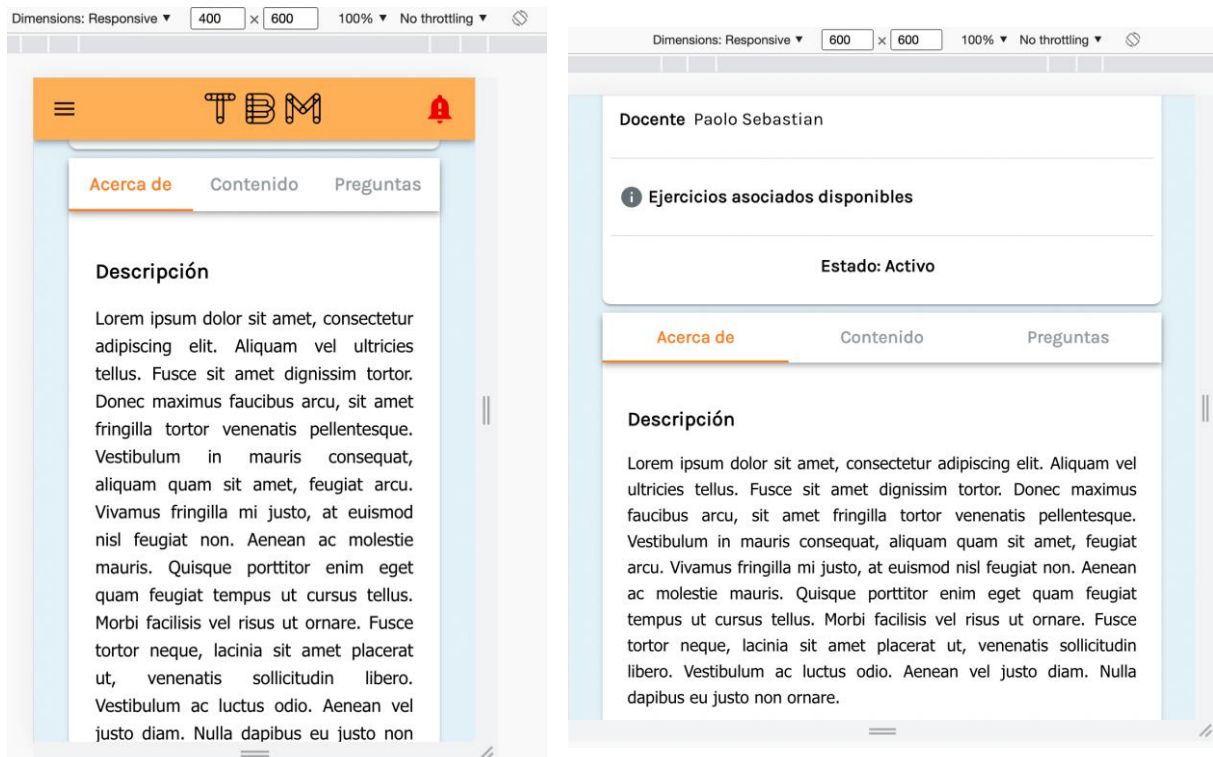


Figura 105 - Vista de tabs en detalle de un curso en 400px (izquierda) y 600px (derecha)

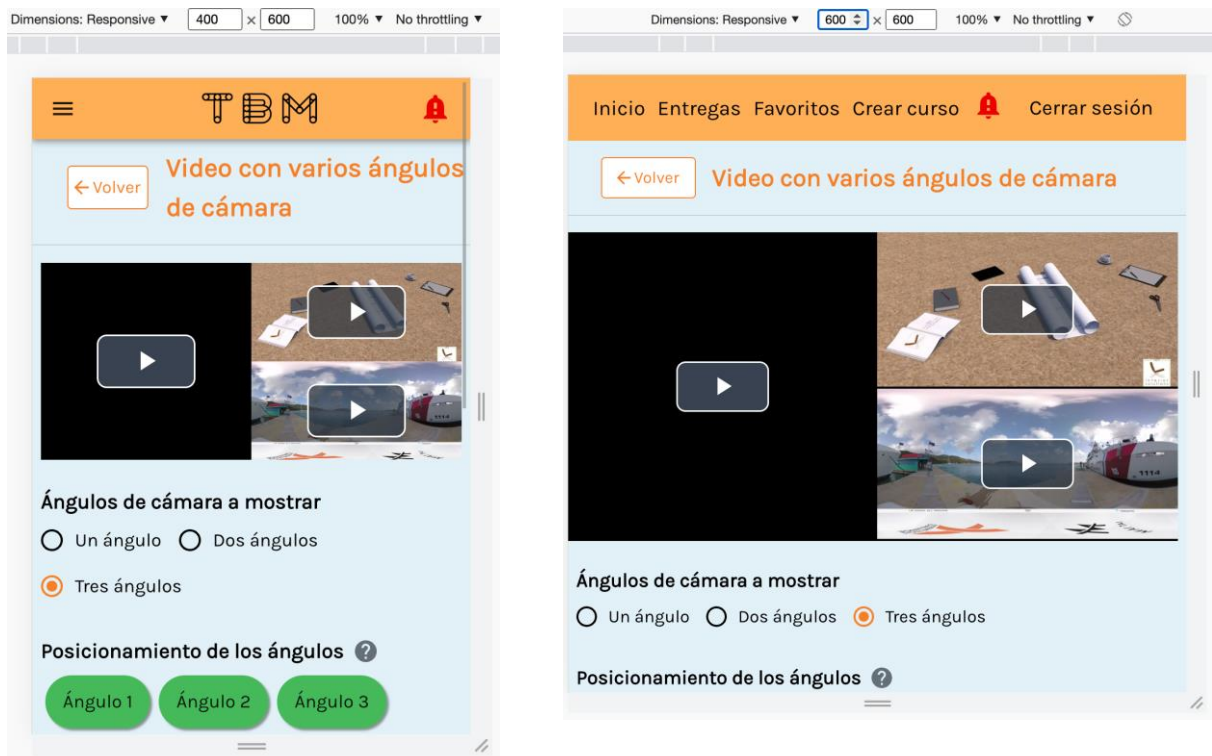


Figura 106 - Vista de reproductor de video de un curso en 400px (izquierda) y 600px (derecha)

13.24. Anexo 24: Cumplimiento de compatibilidad con navegadores (RNF8)

A continuación, se brindará evidencia del cumplimiento del RNF8, en particular, mostrando capturas de la vista en detalle de un curso y la visualización de vídeos con distintos ángulos de cámara.

Para las validaciones, se utilizaron capturas de los navegadores de escritorio, Safari y Google Chrome para macOS. Y para la validación en los dispositivos móviles, se utilizaron emuladores. En el caso de iOS se utilizó el navegador Safari desde un emulador de un iPhone 13 pro con la versión del sistema operativo número 15.5. En el caso de Android se utilizó un emulador de un Pixel 4 con la versión de sistema operativo número 12.

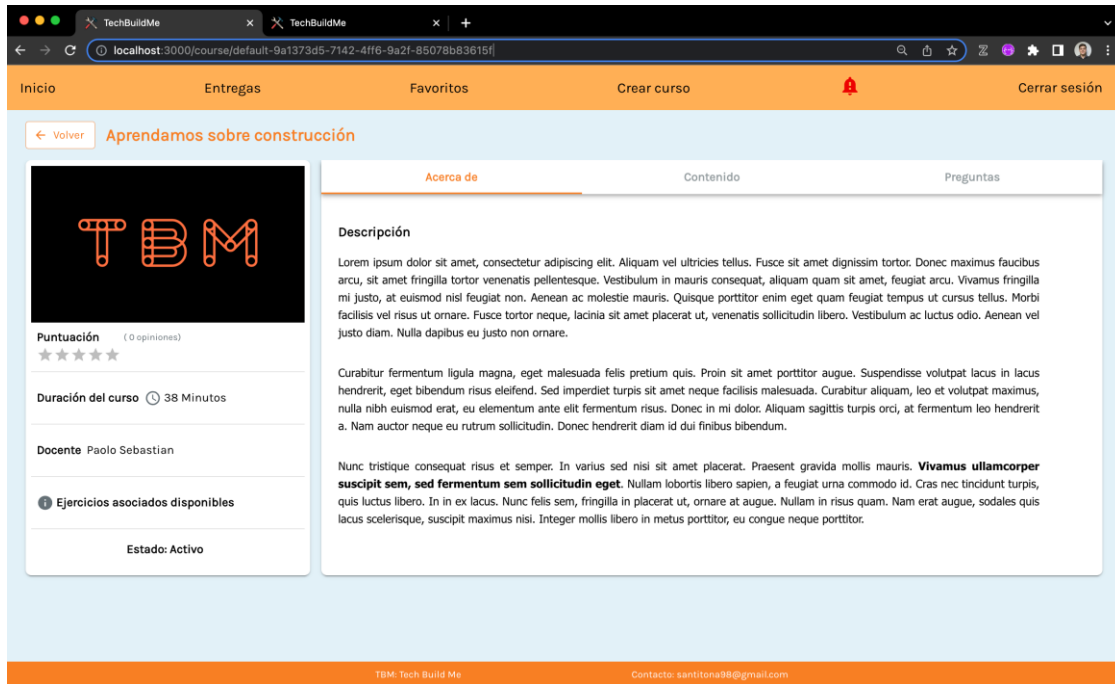


Figura 107 - Vista de detalle de un curso en Google Chrome

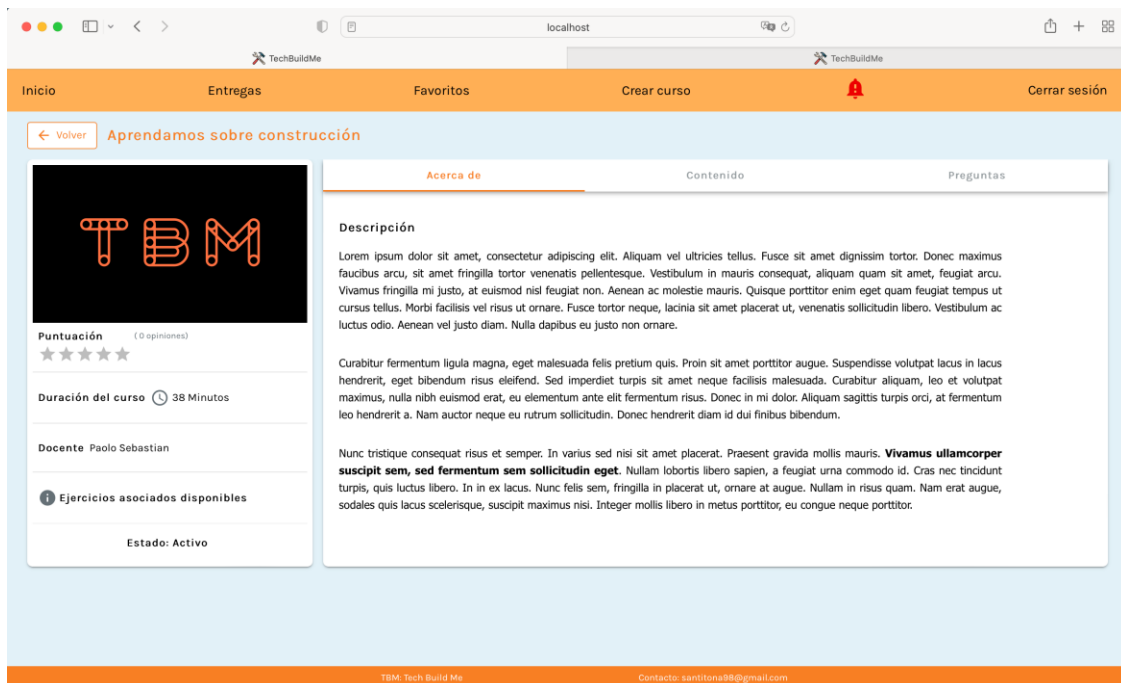


Figura 108 - Vista de detalle de un curso en Safari

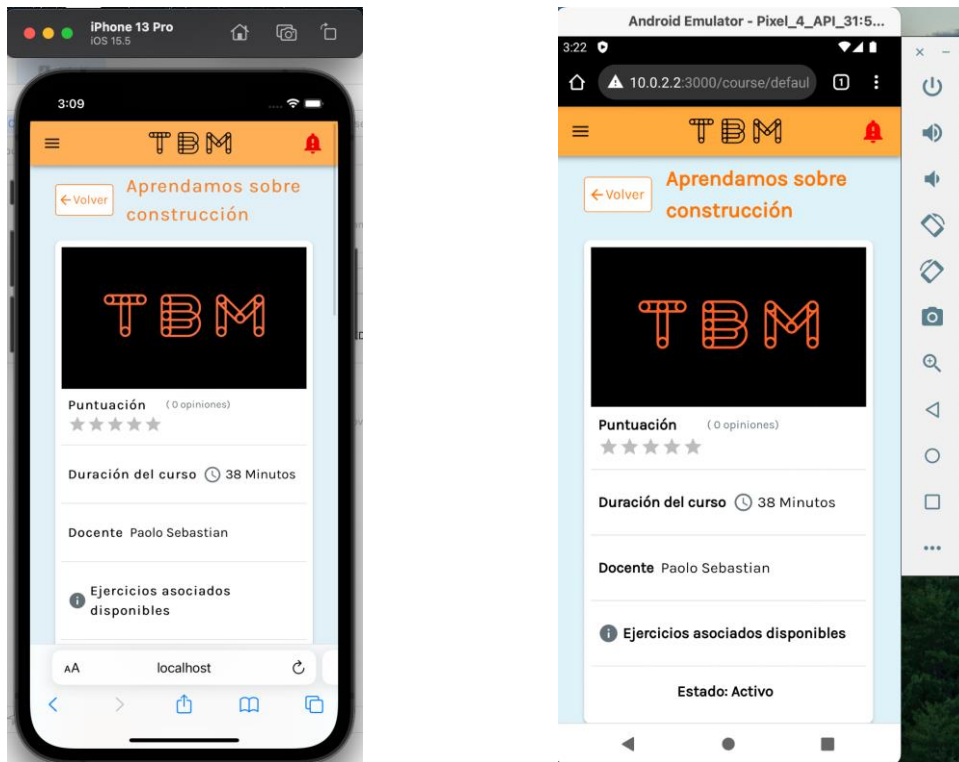


Figura 109 - Vista de detalle de un curso en un iPhone 13 (izquierda) y Android Pixel 4 (derecha)

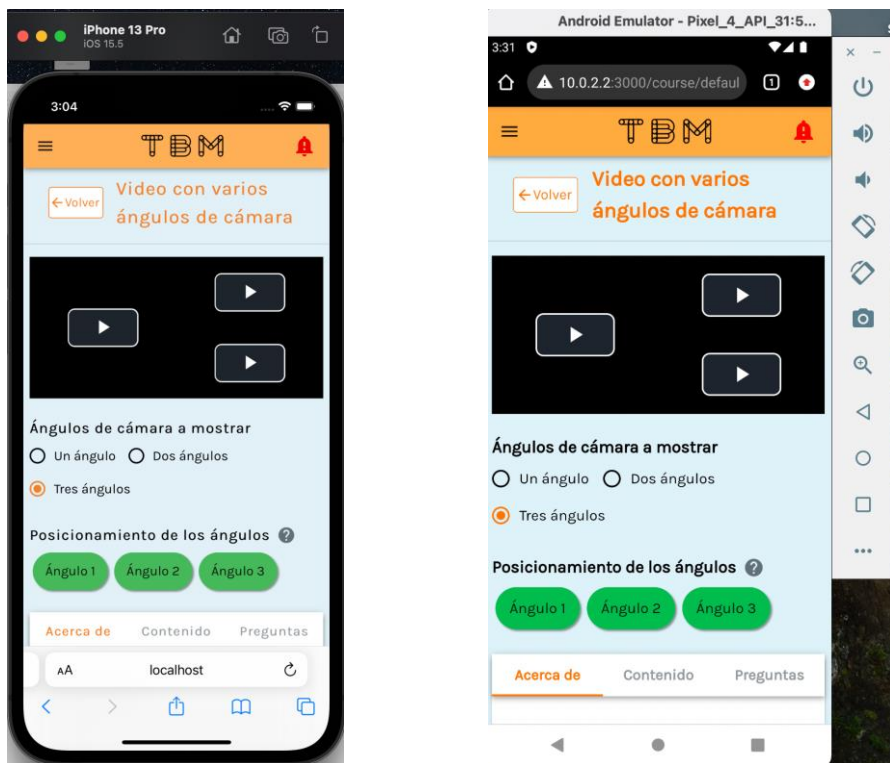


Figura 110 - Vista de reproducción de video de un curso en un iPhone 13 (izquierda) y Android Pixel 13 (derecha)

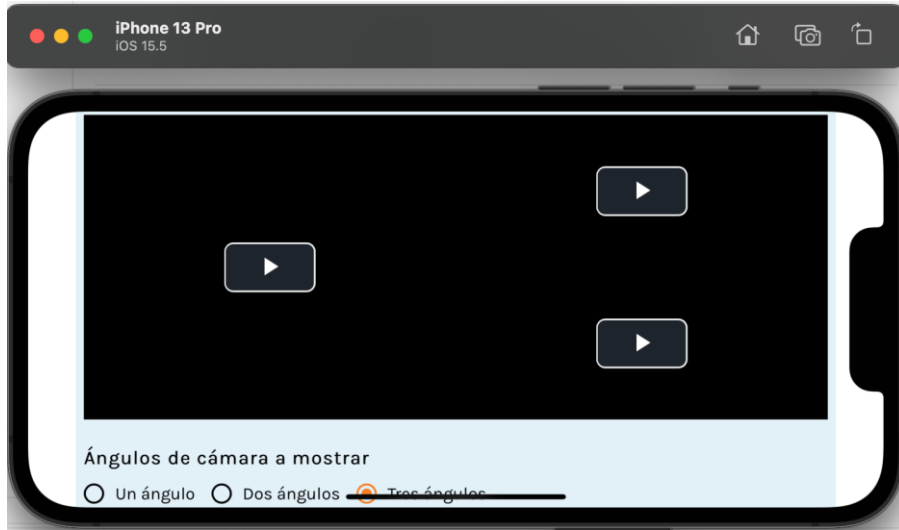


Figura 111 - Vista de reproducción de un curso en un iPhone 13 (arriba) y Android Pixel 4 (abajo) en landscape

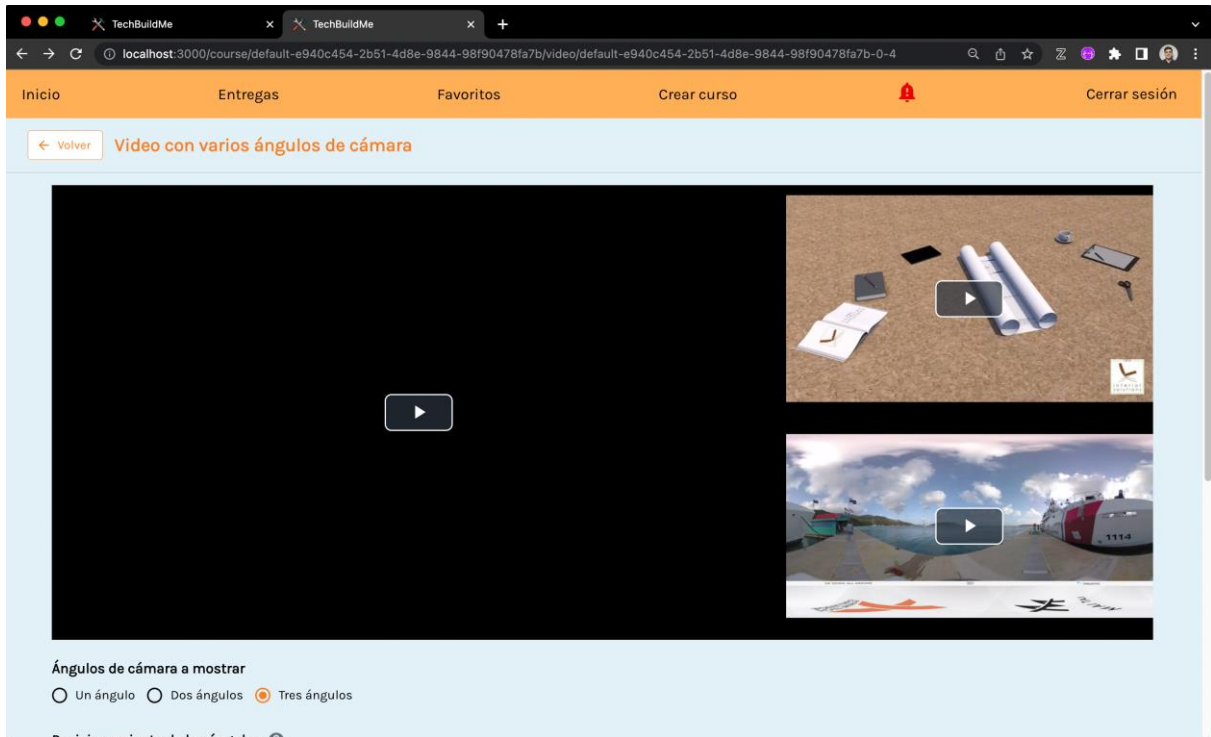


Figura 112 - Vista de reproducción de video de un curso en Google Chrome

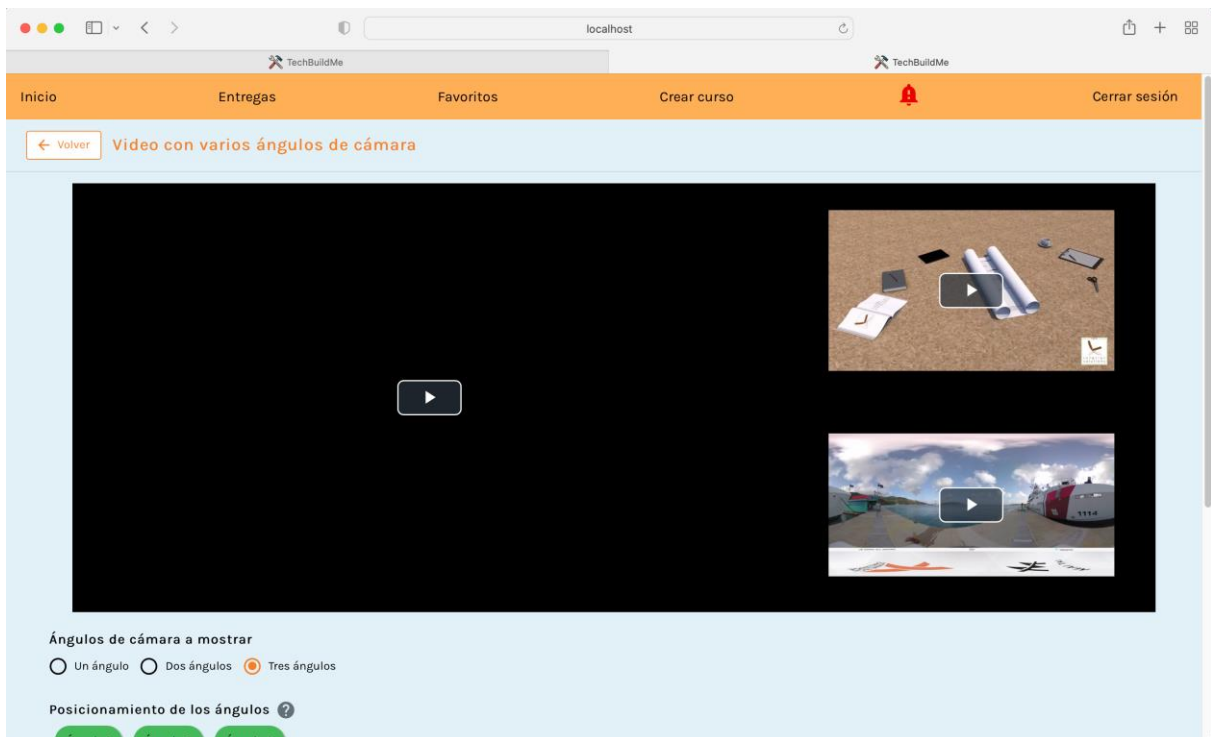


Figura 113 - Vista de reproducción de video de un curso en Safari