



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

GESTIÓN DE PROYECTO PARA LA CREACIÓN DE UN SISTEMA INFORMÁTICO ORIENTADO A LA GESTIÓN ACADÉMICA

Benjamín Ignacio Iturrieta Grondona

Benjamin.iturrieta@usm.cl

Ricardo Cahe
Profesor Guía

Rodrigo Pinochet
Profesor Correferente

2026



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título Tesis de Postgrado

Título del trabajo: GESTIÓN DE PROYECTO PARA LA CREACIÓN DE UN SISTEMA INFORMÁTICO ORIENTADO A LA GESTIÓN ACADÉMICA

Nombre del candidato(a): Benjamín Ignacio Iturrieta Grondona

Carrera / Grado: Ingeniería en Informática

Campus: Sede de Viña del Mar **Departamento:** Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Ricardo José Cahe Cabach, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

6 meses 12 meses 2 años 3 años 5 años 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 2-3-2026 Firma: 

Estudiante o Candidato(a):

Fecha: 02/03/2026 Firma: 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.



Agradecimientos: Quiero agradecer a mi madre, que estuvo siempre ahí en mis peores momentos, me apoyó en todas las caídas que tuve, hasta en mis momentos de desistir por este camino, muchas veces no me sentía capaz en mis primeros años y su forma de animarme a pelear por lo que quiero me permitió llegar a donde estoy; Mi pareja Gabriela Monge que llegó en un momento donde mi carga académica era más alta que nunca, tenía bastantes ramos y debía estar continuamente estudiando o haciendo proyectos con su informe asociado para el desarrollo del semestre, sin sus ánimos y su forma distinta de ver el mundo, capaz de darme siempre una opción fuera de mi campo de visión, como su capacidad de siempre creer en mí, que puede lograr lo que sea que me proponga, me impulsó a sentirme imparables hasta llegar a finalizar mis estudios, a mis amigos Loreto, Magno, Diego y Sebastián, hicieron que este camino fuera siempre entretenido y compartir con ustedes toda esta etapa académica fue un total agrado. Les estaré siempre agradecido por todo lo que han hecho por mí.

Resumen: La tesina aborda la problemática proporcionada por el Cuerpo de Bomberos Viña del Mar (CBVM), que enfrenta un déficit en sus sistemas digitales, los cuales no constan con un sistema capaz de enfrentar las necesidades académicas de su institución. Actualmente son capaces de proporcionar el seguimiento académico a través de un Excel, el cual no permite visualizar todo el contenido necesario para administrar un sistema, por lo que distintas áreas son manejadas a lápiz y papel.

La solución consiste en centralizar la documentación relacionada a la generación de certificados, el monitoreo de cursos, módulos y especialidades, como también la digitalización del personal bomberil de Viña del Mar. Debido a la limitada información nos orientamos por sistemas ya conocidos, y sugeridos por el CBVM. Se utilizó para modelar, herramientas como Angular, Django y MongoDB, usando metodologías ágiles para el desarrollo del proyecto, dividiendo el trabajo a ejecutar en sprints para asegurar la calidad del producto ante el cliente, adaptando los requisitos necesarios.

Se priorizó la estructuración de la aplicación web, como el desarrollo para digitalizar y proporcionar continuidad a la malla curricular bomberil, separando por roles las interfaces de la página, como también proporcionando capacidades de matricular, registrar especialidades, cursos y módulos, asignar instructores, modificar roles en la aplicación y generar prerrequisitos para la continuidad de los módulos que a su vez puedan ser calificados por los respectivos instructores.

Entre los resultados destacan el correcto seguimiento de la malla curricular para el CBVM como también el eficaz contacto de especialistas registrados, siendo llamados a proporcionar servicio como instructores y a su vez participar en incidentes relacionados a su área de conocimiento. Por otro lado, el digitalizar el sistema permitió tanto a los instructores como los estudiantes facilitar la enseñanza y el aprendizaje, considerando que actualmente pueden registrar comunicados a los alumnos y proporcionar material didáctico a los respectivos módulos; un buen control de decisiones ante el cambio apoyado a la refactorización de requisitos. Esto, combinado la utilización de metodologías ágiles, permitió al equipo entregar un producto mínimo viable capaz de solventar las necesidades del cliente, siendo un sistema escalable y con capacidad de mantenerse en el tiempo como una solución útil y eficaz para esta comunidad.

Palabras clave: Sistema de gestión académica, metodologías ágiles, aplicación web, Angular, Django.



Índice

1	Introducción.....	4
1.1	Planteamiento del Problema.....	4
1.2	Solución General	4
1.3	Contexto	4
1.4	Relevancia de las metodologías ágiles en proyectos de software	5
1.5	Descripción del problema	5
1.6	Objetivo general	6
1.7	Objetivos específicos	6
2	Marco Teórico	6
2.1	Principios y valores del manifiesto ágil	6
2.2	Descripción de metodologías ágiles	8
3	Metodología y plan de Trabajo.....	10
3.1	Planificación temporal del proyecto	11
4	Planificación de sprints	12
4.1	Definición de Sprints y ciclos de desarrollo	12
4.2	Estimación de tareas y asignación de recursos.....	13
4.3	Historias de usuario.....	13
4.4	Desarrollo real de proyecto – Roadmap del sistema.....	23
5	Ejecución de reuniones ágiles.....	24
5.1	Descripción de reuniones diarias, de planificación, revisiones y retrospectivas	24
5.2	Importancia de la comunicación y colaboración en el equipo	25
6	Gestión de cambios y mejora continua	26
6.1	Adaptación a cambios en los requerimientos.....	26
6.2	Gestión de mejoras basadas en feedback.....	26
6.3	Tabla de las Historias de usuario a Refinar	27
6.4	Tabla de Gestión del alcance	28
6.5	Tabla gestión de riesgos	30
7	Evaluación del desempeño ágil	30
7.1	Métricas de evaluación	31
7.2	Análisis de resultados	32
8	Desafíos y soluciones	33
9	Estimación de costos	35
10	Conclusiones	36
11	Anexos.....	39



Tabla de Ilustraciones

Ilustración 1 – Backlog de la iteración 2 en formato Kanban.....	9
Ilustración 2 – Roadmap con funcionalidades y tareas de la iteración 1.....	23
Ilustración 3 – Roadmap con funcionalidades de la iteración 2.....	23
Ilustración 4 – Roadmap de las funcionalidades de la iteración 3.....	24

Tabla de Tablas

Tabla 1 - Historias de usuario a refinar.....	27
Tabla 2 - Gestión del alcance.....	28
Tabla 3 - Gestión del alcance.....	29
Tabla 4 - Gestión de Riesgos.....	30
Tabla 5 - Costos a asociados a la estructuración del sistema.....	35
Tabla 6 - Costos asociados a la mantención del sistema posterior al primer año de uso.....	36

1 Introducción

1.1 Planteamiento del Problema

El cuerpo bomberil de Viña del Mar comenzó la digitalización de sus sistemas en 2021, buscando implementar un seguimiento a las certificaciones de los miembros de su institución, que años anteriores al mencionado, no poseían más que un registro físico de la realización de los distintos cursos y entrenamientos, lo que impide el contacto de personal bomberil especializado o el llamado a distintos miembros del cuerpo de bomberos a distintas misiones del área, como también el desarrollo de la academia de bomberos en relación a proporcionar información a su alumnado en tiempo real, como también un seguimiento estadístico, capaz de guiar al profesor en el progreso de las clases, como son el progreso del curso, la asistencia general de su alumnado y los alumnos inscritos en los distintos cursos o especialidades impartidas.

1.2 Solución General

La solución consiste en la creación de una plataforma tecnológica centralizada que articule a la Academia de Bomberos y al Cuerpo de Bomberos Viña del Mar (CBVM), centralizando la documentación relacionada a la generación de certificados, el monitoreo de cursos, módulos y especialidades, la digitalización del personal bomberil de Viña del Mar.

1.3 Contexto

Durante el primer semestre, nuestro equipo tuvo que escoger un desafío propuesto por un patrocinador, entre distintas entidades en las cuales el CBVM, presentó un proyecto asociado a desarrollar una plataforma tecnológica centralizada que articule al CBVM y a la Academia Bomberil, permitiendo digitalizar certificados y monitorear la formación bomberil para las distintas compañías de Viña del Mar, asegurando la trazabilidad curricular y fomentando el desarrollo profesional permanente.



Entre las historias de usuario destacadas para realizar este proyecto están:

- 1.- Como administrador quiero definir cursos con prerequisites para controlar la progresión académica y garantizar el cumplimiento de la malla curricular
- 2.- Como rector académico de Bomberos quiero un sistema que genere certificados de mis alumnos para que lleven constancia de todos los cursos que han realizado
- 3.- Como Alumno estudiante de bomberos quiero revisar los cursos que me faltan por hacer para llevar seguimiento de mi carrera bomberil
- 4.- Como Rector Académico de bomberos quiero un sistema de filtrado para reconocer cuantos bomberos poseen las distintas especialidades del CBVM
- 5.- Como personal administrativo de bomberos quiero poder registrar a los bomberos que no pudieron anexarse en el sistema pre-2021 para llevar constancia de los cursos que realizaron y el grado en el cual están actualmente.

1.4 Relevancia de las metodologías ágiles en proyectos de software

Las metodologías de categoría “Ágil”, propician un desarrollo fundamental en proyectos de desarrollo de Software, entre estas razones es posible mencionar:

- **Enfoque de mejora continua:** El fin de cada iteración permite al equipo desarrollador reconocer sus falencias tanto como aciertos, lo que permite buscar soluciones ante obstáculos que detecto el equipo en el desarrollo del proyecto.
- **Adaptabilidad a los cambios:** Los equipos que utilizan metodologías ágiles priorizan la entrega de valor ante sistemas que no cumplan con los criterios del cliente, es por esto, que adaptar a las necesidades del usuario el modelo de software es vital para crear valor en el producto.
- **Comunicación constante con el equipo de desarrollo:** Comunicarse entre todos los miembros del equipo es vital para detectar obstáculos, adaptarse a los cambios y desarrollar a un ritmo continuo la implementación de funcionalidades.
- **Reportes iterativos con los interesados del proyecto:** Integrar al cliente a las distintas fases del desarrollo de software permite detectar las necesidades del cliente conforme se crean funcionalidades.
- **Reducción de riesgos:** Al reportar al cliente las distintas implementaciones, reduce significativamente los riesgos de cambiar funcionalidades grandes en el sistema.
- **Priorización del usuario:** Las metodologías ágiles se centran en la satisfacción del cliente para entregar valor al producto final.

1.5 Descripción del problema

La escuela de maestría bomberil de Viña del Mar es el organismo encargado de establecer y supervisar los estándares de capacitación y entrenamiento para todos los cuerpos de bomberos de la comuna. Se encargan de desarrollar una malla curricular con todos los niveles y áreas de formación, asegurando que los bomberos adquieran las competencias necesarias para desempeñar sus funciones de manera eficiente. Actualmente debido a la obsolescencia de sus sistemas, poseen dificultades a la hora de manejar la documentación relacionada a los distintos miembros del CBVM. Esto, complementado a la escasez de sistemas digitales por parte del cuerpo bomberil, esto lleva a no disponer de un seguimiento total de los cursos ni módulos completados por los miembros de las distintas compañías. El CBVM dispone meramente de un sistema no escalable a partir de 2021 hacia la actualidad, prescindiendo de un registro práctico para la inscripción de instructores como del alumnado, actualmente todo el sistema es manejado a nivel físico, todo es registrado en libros y planillas para llevar un nivel de seguimiento de asistencia, calificaciones, continuidad de la malla



curricular, como los instructores capaces de dictaminar en los respectivos módulos, lo que impide mantener un seguimiento en el desarrollo de la malla académica para el alumnado, consiguiendo el dictamen de módulos a repartir en el semestre académico, como también el seguimiento de especialidades completadas a causa que los certificados no están siendo respaldados en un sistema virtualizado, y no hay un sistema eficaz para corresponder el nivel de demanda de los miembros del cuerpo de Viña del Mar, dificultando el contacto de especialistas a la hora del combate de incendios u otro tipo de incidentes relacionados, influyendo de igual manera al próximo desarrollo de la malla académica, orientado a próximas especialidades.

1.6 Objetivo general

Gestionar un sistema informático, centralizado de gestión académica a través de un enfoque de Gestión de proyecto, utilizando metodologías ágiles basadas en Scrum y Kanban, alineando las necesidades del CBVM y la escuela de maestría bomberil, asegurando la entrega de un sistema escalable, seguro y mantenible en el tiempo.

1.7 Objetivos específicos

- Organizar los distintos requisitos del cliente, definiendo objetivos claros, realistas y abordables según las capacidades del equipo desarrollador.
- Dirigir, planificar, organizar y controlar el desarrollo de software del proyecto bajo un marco ágil, mediante herramientas asociadas a la gestión de proyecto.
- Realizar reuniones con el equipo desarrollador para el cumplimiento de los estándares establecidos por el grupo proyecto, con la combinación de roles de Scrum Master y Product Owner.
- Validar la elaboración de diseños y vistas para la implementación de interfaces acordes al alcance del proyecto.
- Comunicar a los interesados el avance del proyecto mediante la elaboración de reuniones y *demos*.
- Mantener la cantidad de funcionalidades acorde a la capacidad de desarrollo de los programadores.

2 Marco Teórico

2.1 Principios y valores del manifiesto ágil

Creado en febrero del 2001 en EE.UU, es un documento fundacional que, para su época, revolucionó la forma en que se desarrollan proyectos y productos en el ámbito de la tecnología. El Manifiesto Ágil, establece valores y principios fundamentales que guían una metodología de desarrollo.

Valores del Manifiesto Ágil

- **Individuos e interacciones sobre procesos y herramientas:** Las personas y sus interacciones son vitales para el éxito de un proyecto, debido a su capacidad de ser creativas e innovar, mediante la propuesta al cambio u la adaptación, mientras, que las herramientas son un apoyo para que los individuos cumplan sus objetivos, como también los procesos son una ayuda para guiar el trabajo.



- **Software funcionando sobre documentación completa:** La documentación exhaustiva de requisitos, trasladan al cliente a un conjunto de funcionalidades por realizar, para remediar tal situación es necesario priorizar la interacción con el cliente en etapas tempranas, buscando una retroalimentación en los distintos prototipados del sistema, lo que pondrá a los usuarios a tomar decisiones sobre el producto, y ayudará al equipo desarrollador a tener una visión clara de cuales funcionalidades adaptar o cambiar.
- **Colaboración del cliente sobre la negociación del contrato:** El cumplimiento estricto de un contrato y mantener los requisitos iniciales para la elaboración de un sistema, no es una táctica oportuna, dado que esto limita la entrega de valor y crea barreras con los interesados, por este motivo, es necesario integrar al cliente como un miembro más del equipo, a través de su colaboración continua, es posible obtener una retroalimentación en todo el transcurso del producto a desarrollar, lo que aumenta el valor final del producto.
- **Respuesta ante el cambio por encima de seguir un plan:** En escenarios inestables, la planificación y control que proponen las fórmulas de gestión tradicionales, no son acordes a un entorno de desarrollo de software, la capacidad de respuesta y adaptación que promueven las metodologías ágiles, en cambio, son una ventaja competitiva para las organizaciones.

Principios del Manifiesto Ágil

- **Satisfacer al cliente mediante la entrega temprana y continua:** La entrega pronta y frecuente de software permiten mantener como foco la satisfacción del cliente, lo que favorece desde etapas tempranas, reducir la posibilidad de realizar cambios significativos en fases cercanas al finalizar el incremento.
- **Aprovechar el cambio como ventaja competitiva:** Aceptar y atender las modificaciones solicitadas por el cliente, reconociendo el aumento de valor que aportan los cambios a implementar, proporcionen una relevancia en el producto y aporten una ventaja competitiva al atender las necesidades actuales del usuario.
- **Entregar valor frecuentemente:** Proporcionar actualizaciones de software en base a subproductos, permite reducir la probabilidad que el cliente demande cambios mayores del producto posteriormente, mientras más retroalimentación por parte del interesado, menor es la probabilidad de producir errores en el desarrollo.
- **Cooperación negocio-desarrolladores durante todo el proceso:** Los equipos encargados de desarrollo y del negocio deben colaborar entre sí, abordando malentendidos, compartiendo retroinformación mutuamente y alineando sus intereses, dejando de lado las barreras existentes.
- **Construir proyectos en torno a individuos motivados:** Motivar y propiciar un espacio seguro para el equipo desarrollador, crea un ambiente de confianza para participar en la toma de decisiones relacionadas a las fases de desarrollo de software, buscando que se sientan identificados con el propósito del proyecto y entreguen su compromiso a desarrollar un producto de calidad.
- **Utilizar la comunicación cara a cara:** La comunicación cara a cara, tanto en medios presenciales como remotos, son una solución a tiempos de respuesta prolongados y malentendidos comunes a la hora de la utilización de medios como son los hilos de correo electrónico.
- **Software funcionando como medida de progreso:** Acciones que no repercuten en la creación de un producto, no permiten demostrar un avance verdadero en un proyecto, es por esto, que entregar funcionalidades mide la cantidad de progreso realizado a la hora de desarrollar software, en cuanto se habla de la vista de un cliente.
- **Promover y mantener un desarrollo sostenible:** Mantener un ritmo que pueda ser seguido por todos es fundamental para evitar esfuerzos desmesurados, presiones excesivas y tensiones en el equipo, generar una forma óptima de trabajo, permite entregar al mercado soluciones de software que aporten valor a los usuarios de manera frecuente.

- **La excelencia técnica mejora la agilidad:** Cuidar aspectos técnicos, como un buen diseño lógico, a la hora de desarrollar un producto de software, mejora la calidad de mejorar el producto cuando nuevas necesidades vayan surgiendo.
- **La simplicidad es fundamental:** Es requerido priorizar las acciones que realmente aporten valor, por lo que es necesario actuar de la forma más sencilla posible, simplificando tareas como la automatizar procesos, eliminar funciones innecesarias y utilizar librerías de software.
- **Equipos autogestionados para generar más valor:** Propiciar un nivel de confianza suficiente y un grado de libertad al grupo desarrollador, evitando un control excesivo sobre estos, permite la autogestión y mejores resultados a la hora de desarrollar
- **Reflexión y ajustes frecuentes del trabajo de los equipos:** La reflexión frecuente sobre el trabajo realizado, permite al equipo ajustar y mejorar su rendimiento, buscando una mejora continua y no conformarse con cumplir con lo solicitado.

2.2 Descripción de metodologías ágiles

Scrum

Scrum es una metodología de trabajo ágil que se centra en estructurar y gestionar el trabajo mediante un conjunto de valores, principios y prácticas. El método scrum anima al equipo a aprender a través de la experiencia, a la autogestión mientras abordan un problema, reflexionar sobre sus aciertos y sus errores para mejorar continuamente. La manera de repartir una problemática se realiza mediante iteraciones cortas, de más de un mes llamadas “*Sprints*”, que pueden durar de 2 a 4 semanas.

Roles:

- **Scrum Master:** Lidera el equipo de Scrum y mantiene a los miembros enfocados en los principios del marco de trabajo, ayuda a gestionar el trabajo pendiente del producto, organiza reuniones de planificación del sprint como reuniones diarias, realiza análisis retrospectivos y elimina obstáculos para el equipo.
- **Product Owner:** Transforma los requisitos del cliente en características de trabajo pendiente del producto (Product Backlog) para que el equipo Scrum trabaje en ellas, actúa como nexo entre las partes interesadas del negocio, los miembros del equipo de Scrum y los usuarios finales.
- **Development Team:** Son el equipo encargado de desarrollar las funcionalidades descritas en el trabajo pendiente del producto.

Eventos:

- **Sprint:** Iteración del trabajo a realizar con la finalidad de alcanzar un incremento de producto utilizable y desplegable.
- **Reunión de planificación del Sprint:** Define el objetivo del Sprint (Sprint Goal) y el trabajo a realizar en la duración de la iteración.
- **Scrum diarios:** Reuniones operativas, su objetivo es detectar obstáculos que impiden el avance del desarrollo del proyecto y ayuda a la toma de decisiones que favorezcan el cumplimiento de los tiempos a la consecuencia del “*Sprint Goal*”.
- **Revisión del Sprint:** Reunión informal, se realiza al finalizar una iteración, su objetivo es a través de las partes interesadas, recopilar comentarios sobre el trabajo realizado y optimizar el *Product Backlog*.
- **Retrospectiva del Sprint:** Reunión táctica, se centra en analizar los objetivos alcanzados, tomando en cuenta los aciertos y los fallos del equipo, buscando solucionar las problemáticas detectadas para mejorar para la siguiente iteración.
- **Perfeccionamiento (Refactorización):** Reunión realizada entre la realización del Sprint, permite rediseñar, priorizar, modificar y simplificar las tareas introducidas en el *Product Backlog* de la iteración.

Artefactos:

- **Backlog del Producto:** Lista priorizada de los requisitos transformados por el Product Owner en tareas realizables para desarrollar el producto.
- **Backlog del Sprint:** Tareas a completar en el sprint, contiene el Sprint Goal como las historias de usuario seleccionadas del Backlog del Producto.
- **Incremento:** Resultado del trabajo completado, medible en funcionalidades entregadas al cliente.

Kanban

La subcategoría de gestión ágil Kanban, es orientada a implementación visual de la gestión de proyectos, esta permite visualizar las distintas etapas del flujo de trabajo, enlistando las tareas por realizar, y el estado del requisito hasta ser completado.

Elementos Clave

- **Tablero Kanban:** Representa el flujo de trabajo con columnas que muestran las distintas etapas del trabajo en curso, estás poseen un límite de tareas en cada etapa, para no colapsar las distintas instancias de los entregables.
- **Tarjetas Kanban:** Son las representaciones móviles de las tareas en el tablero, cada tarjeta es una tarea por realizar, lo que detalla un requisito por cumplir, están colocadas en distintas columnas mostrando el estado exacto de realización.
- **Software Kanban:** Los miembros del equipo pueden utilizar el tablero Kanban, editando las tareas y modificando el estado de las distintas tareas de la iteración.

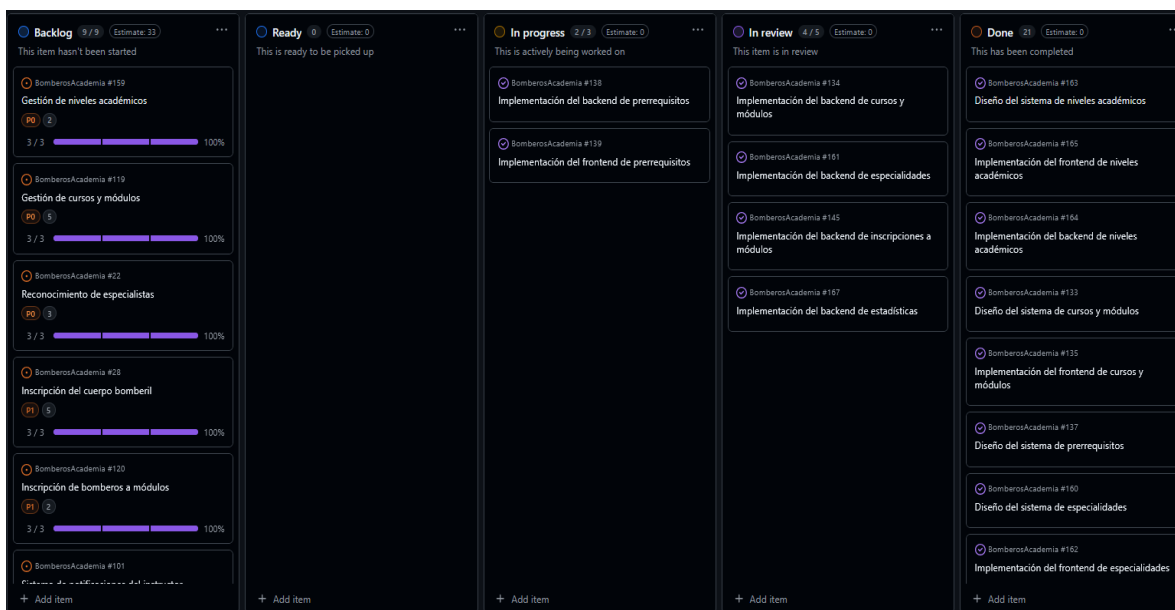


Ilustración 1 – Backlog de la iteración 2 en formato Kanban

3 Metodología y plan de Trabajo

Para desarrollar este proyecto será necesario priorizar un mayor número de desarrolladores, esto relacionado a funcionalidades extensas que son de alta prioridad para llevar a cabo este sistema, por lo que serán combinados los roles de Scrum Master y Product Owner, que a pesar de la discordancia entre objetivos, referido al método para tomar decisiones (El Scrum Master prioriza decisiones a nivel estratégico y el Product Owner prioriza decisiones a nivel táctico) y a los enfoques que deben priorizar ante los desarrolladores, permitirá al equipo desarrollador coordinar e implementar las distintas funcionalidades a repartir en las distintas iteraciones del sistema, distribuyendo roles como Arquitecto de Software, Desarrollador Front-End, Desarrollador Back-End y Desarrollador Cloud con vistas a automatización. Mientras que el Scrum Master-Product Owner se encarga de liderar al equipo en todas las tareas relacionadas al ámbito de gestión como también la comunicación con el cliente, además de velar por un desarrollo de software orientado a las necesidades de la institución interesada en el proyecto.

Para el desarrollo óptimo de este proyecto, se utilizará la herramienta de GitHub orientada a la gestión de proyectos, herramienta utilizada para tableros Kanban e implementar la metodología ágil Scrum, como también documentar información relacionada a épicas, historias de usuario, asignación de tareas, estimación de historias de usuario, la notificación de bugs y los fix necesarios en el avance de cada iteración.

También la plataforma de GitHub será utilizada para manejar los repositorios de código asociados a los incrementos, además de estar relacionadas conforme al avance de cada funcionalidad y requisitos del sistema, siendo manejada con una rama principal “main” utilizada para mostrar el producto a los interesados del proyecto; la rama orientada al desarrollo, siendo además un paso anterior antes de llevar las funcionalidades a la rama principal “develop”, que será un área integración de las funcionalidades desarrolladas, como también utilizada para verificar los criterios necesarios para ser integradas posteriormente en main; la rama orientada al desarrollo de funcionalidades específicas “feature”, manteniendo nombres clave relacionado a la funcionalidad a desarrollar, y la rama de “hotfix”, manteniendo un nombre clave a la hora de ser integrada en “develop”, manteniendo el nombre del autor, el sprint y el nombre clave “fix” en su terminología.

La metodología por utilizar es una combinación de Scrum con Kanban, los sprints, reuniones de retrospectiva, de refactorización, planificación, revisión, son los principales ejes a trabajar de scrum, en cuanto a las reuniones diarias, debido a la incompatibilidad horaria y a las altas horas de traslado de los miembros del equipo, las reuniones diarias no son una prioridad a realizar al pie de la letra, por lo que se mantendrá un mínimo de 3 sprint daily a la semana, en cuanto a Kanban será integrado a la herramienta de GitHub, utilizado para priorizar y realizar las historias de usuario correspondientes a la iteración, mostrar tanto a nivel visual los puntos de historia como el desarrollador encargado de implementar la funcionalidad o componente específico y el estado correspondiente de la implementación.

Debido a que el equipo ha cambiado en dos ocasiones de proyecto esto relacionado a inconvenientes de fuerza mayor, nuestras tareas tanto de planificación como de desarrollo estarán desplegadas solo al segundo semestre académico, lo que lleva a tomar decisiones en su mayoría a nivel táctico y operativo, buscando desarrollar al mismo nivel que vamos planificando la siguiente funcionalidad a integrar al sistema, además de localizar al cliente en base a sus acotados tiempos para alcanzar un nivel de feedback mínimo como también sugerencias y necesidades localizadas en reuniones online.



3.1 Planificación temporal del proyecto

Esta fase convendrá una combinación de las actividades de nuestro primer semestre académico, orientado a realizar la planificación correspondiente a una solución práctica para nuestro cliente, como el diseño de interfaces usuarias, como también la investigación necesaria para desarrollar una arquitectura de software, además del desarrollo de software necesario para abarcar las funcionalidades prioritarias para entregar un producto mínimo viable (MVP) acorde a la dolencia de nuestro patrocinador; esto con el afán de trabajar sin una deuda técnica demasiado extensa, que nos impida poder entregar una solución acorde a los requisitos registrados, a partir del contexto relacionado a la problemática

hemos recolectado los requisitos más importantes, en base a esto se ha generado para el proyecto la *Definition of Done (DoD)*, la cual consta de un acuerdo de buenas prácticas para garantizar la calidad de las implementaciones:

- La solución debe tener un sistema de autenticación para los miembros del CBVM
- La solución debe permitir a los usuarios del sistema descargar sus certificados relacionados al cumplimiento en la progresión de la malla curricular
- La solución debe permitir matricular alumnos e instructores, simulando a todo el cuerpo bomberil
- La solución debe permitir a los instructores evaluar al alumnado en los distintos módulos que imparta este según el curso y nivel que correspondan
- La solución debe permitir descargar los entrenamientos realizados por los bomberos de Viña del Mar
- La solución debe permitir emular toda la malla curricular de Viña del Mar, además de permitir la escalabilidad conforme esta vaya creciendo o cambiando.
- El sistema debe permitir reconocer a los miembros con especialidades ya realizadas, como también miembros del cuerpo en los distintos niveles relacionados a la malla académica
- El sistema debe permitir a los instructores marcar la asistencia de los alumnos
- El sistema debe poseer un sistema de comunicados tanto para los instructores como para el personal administrativo
- El sistema debe permitir filtrar tanto como alumnos como instructores en sus respectivos niveles
- El sistema debe respaldar todos los archivos en un sistema virtual
- El sistema debe permitir asignar instructores para los distintos cursos, como también asignar a un instructor particular para cada módulo
- Los administradores deben poseer las facultades de registrar niveles, cursos módulos y especialidades
- Los administradores deben poseer la capacidad de subir los certificados para los distintos miembros del cuerpo de bomberos
- Los administradores deben poseer la capacidad de inscribir al alumnado en los módulos que correspondan
- El sistema debe poseer la capacidad de negar el acceso a funcionalidades vinculadas a la administración como a los instructores

Cada uno de estos requerimientos serán destinados a ser implementados entre los distintos miembros del equipo desarrollador, los cuales se dividirán las tareas según su área de conocimiento, mientras realizamos los requisitos del cliente, desarrollaremos entornos de prueba para generar las correspondientes correcciones, ya sean pruebas unitarias, como también comunicar el respectivo desarrollo de software al cliente, buscando optimizar las interfaces desarrolladas como también generar cambios correspondientes según la validación de nuestro *Stakeholder*.

4 Planificación de sprints

El proceso de desarrollo de la solución se llevó a cabo utilizando Scrum como base, dividido en 4 iteraciones, siendo el sprint 0 una iteración para la planificación, diseño de interfaces y del modelo de negocios. Debido a la reasignación del proyecto en nuestro segundo semestre académico, estas instancias fueron re hechas en el sprint 1, por lo cual, a la hora de desarrollar y estructurar el proyecto, nos quedamos con 3 iteraciones de un mes, comenzando desde agosto 4 hasta octubre 24, a lo largo de estos 3 meses el equipo se enfoca en escalar la solución con funcionalidades clave, que serán validadas por el cliente y en caso de ser demasiado complejas refactorizadas para separar la Historia de Usuario en tareas abordables por los distintos desarrolladores, además de ser refinadas por el equipo mediante pruebas manuales y automatizadas.

4.1 Definición de Sprints y ciclos de desarrollo

Repartición de los Sprints

Sprint 1:

- **4 – 29 de Agosto:** Implementación interfaces básicas para un sistema de gestión académica, orientadas a un dashboard con información general del usuario de prueba, diseño del perfil del usuario, método de acceso, descarga de los certificados y los entrenamientos realizados, sistema de validación de usuarios.

Sprint 2:

- **30 de Agosto – 26 de Septiembre:** Desarrollo de roles para el sistema, como también funcionalidades repartidas para cada rol, se generan jerarquías para el funcionamiento del sistema, como también gestores para administradores como para instructores, permitiendo a los administradores matricular, emular la malla académica del cuerpo bomberil de Viña del Mar, generar anuncios, como subir certificados; a los instructores se les dio la facultad ejercer los cursos y módulos como también habilitar métodos de comunicación con sus alumnos. Para los alumnos se mejoró la lógica de las vistas disponibles para ser notificados por administradores como instructores, como la visualización de sus módulos; desarrollo de un sistema de notificaciones orientado al instructor y al alumno.

Sprint 3:

- **27 de Septiembre – 24 de Octubre:** Reestructuración a la capacidad de creación del instructor, como también mejoras a la lógica de los gestores administrativos, se integran funcionalidades al instructor para proporcionar continuidad en el desarrollo de módulos y cursos como la posibilidad de proporcionar material didáctico; Modificación a la interfaz del dashboard de alumno, asimilando la vista a contenido generalizado, relacionado con su progreso académico como información relativa a su persona; integración del estado de sus módulos con mejoras técnicas para la obtención de certificados y entrenamientos; Incorporación y de herramientas administrativas relacionadas a la gestión de usuarios, modificaciones visuales para la inscripción de instructores a los módulos respectivos, de la misma manera, se han otorgado mejoras en los gestores relacionados a la inscripción del alumnado a los módulos y para la subida de archivos relacionada a certificados.

4.2 Estimación de tareas y asignación de recursos

Bajo el desarrollo de la metodología Scrum se han detectado 20 funcionalidades, traducidas en los requisitos del sistema, de los cuales 8 son prioritarios para cumplir con el MVP solicitado por el patrocinador.

La estimación de los requisitos del sistema fue separada en tareas más pequeñas, asegurando entregar la funcionalidad por vistas, en vez de interfaces que sean aplazadas a la siguiente iteración o posponer la función hacia un MVP nuevo, para generar una estimación de estas tareas el equipo separó cada Historia de Usuario considerando aspectos técnicos (Interfaz usuaria, diseño, modelado de datos, Front-End, Back-End).

La prioridad de las historias fue pensada en base a los requisitos mínimos para entregar un sistema, que en caso de no poder desarrollarla en tiempos sugerentes se tomaría un camino alternativo, buscando mantener un sistema capaz de cumplir con las principales dolencias de nuestro cliente, desplazando las funcionalidades a desarrollar hacia la siguiente iteración, requisitos relacionadas que aseguren la correcta gestión de usuarios como también una interfaz de usuario atractiva e intuitiva para los distintos miembros del cuerpo bomberil.

4.3 Historias de usuario

Sprint 1:

a) Autenticación de usuarios (5 Puntos de historia)

Como usuario registrado, quiero iniciar sesión con mi correo y contraseña, para acceder de forma segura a mi cuenta y a las funcionalidades del sistema.

- Horas de trabajo: 72 horas.
- Tareas:
 - Diseñar pantalla de inicio de sesión.
 - Implementar formulario de login.
 - Configurar sistema de autenticación.
 - Crear manejo de sesiones y tokens de acceso.
- Criterios de aceptación:
 - Dado que el usuario va a ingresar al sistema, cuando el usuario utiliza credenciales correctas, entonces entra al sistema.
 - Dado que el usuario requiere entrar al sistema, cuando el usuario utiliza credenciales erróneas, entonces recibe un error de “Usuario o contraseña incorrecta”.
 - Dado el usuario requiere entrar, cuando el número de intentos supera las 5 veces al ingresar con credenciales erróneas, entonces la cuenta queda bloqueada temporalmente.
- Pruebas unitarias
 - Inicio de sesión con credenciales válidas
 - Rechazo de credenciales incorrectas
 - Bloqueo de cuenta por intentos fallidos

b) Dashboard y perfil del usuario (3 Puntos de historia)

Como usuario autenticado, quiero acceder a un panel que me muestre información relevante y accesos rápidos, para navegar fácilmente por el sistema y consultar datos importantes.

- Horas de trabajo: 72 horas.
- Tareas:
 - Implementar sección de perfil de usuario.
 - Diseñar estructura del dashboard.



- Integrar accesos rápidos a funcionalidades principales.
 - Crear widgets informativos.
- Criterios de aceptación:
 - Dado que entró al sistema, cuando va al panel de usuario, entonces ve accesos según su rol
 - Dado que es alumno, e ingresó al sistema, entonces en el dashboard ve accesos a “Mis módulos, certificados y entrenamientos”
 - Dado que es instructor, al ingresar al sistema ve en el dashboard los módulos a impartir, entonces verá los avisos que ha creado, y también estadísticas relacionadas a los módulos que imparte.
- Pruebas unitarias
 - Visualización de accesos según rol del alumno
 - Visualización de accesos según rol del instructor
 - Restricción de acceso sin autenticación
- c) Certificaciones de la malla curricular (8 Puntos de historia)**

Como Rector Académico de Bomberos quiero un sistema que permita acceder a los certificados de mis alumnos para que lleven constancia de todos los cursos que han realizado.
- Horas de trabajo: 117 horas.
- Tareas:
 - Implementar descarga en formato PDF.
 - Crear historial de certificados emitidos.
 - Implementar sistema de búsqueda y listado.
 - Desarrollar visualización de certificados existentes.
 - Diseñar un sistema de almacenamiento de certificados.
- Criterios de aceptación:
 - Dado que es miembro del CBVM, cuando entre al perfil entonces podrá ingresar al apartado de las certificaciones apiladas por el grado completado o a realizar.
 - Dado que un alumno completó un curso, cuando un administrador suba el certificado asociado al alumno, entonces éste podrá ver en su perfil reflejado el curso completado con un archivo a descargar.
 - Dado que un administrador subió un certificado, entonces el gestor administrativo asociado registrará que un usuario lo subió y cuando fue subido.
- Pruebas unitarias
 - Visualización de certificados por alumno
 - Carga de certificado por administrador
 - Registro de auditoría al subir certificado
- d) Entrenamientos del usuario (3 Puntos de historia)**

Como Alumno Estudiante de Bomberos quiero tener un registro de mis entrenamientos para llevar constancia de la fecha y el tipo de entrenamiento en el cual participé.
- Horas de trabajo: 90 horas.
- Tareas:
 - Diseñar interfaz de registro de entrenamientos.
 - Crear historial cronológico de entrenamientos.
 - Crear formulario de carga de entrenamientos.
- Criterios de aceptación:
 - Dado que al ingresar al perfil en la sección de entrenamientos no tenga ningún entrenamiento asociado, entonces se entrega un mensaje de “Aún no registra entrenamientos”.



- Dado que es miembro del CBVM, cuando ingresa al perfil en la sección de entrenamientos entonces verá un listado con título, fecha, descripción de la actividad y un certificado del entrenamiento en el cual participó.
- Dado que quiera obtener un listado de todos los entrenamientos realizados, cuando ingrese en la sección de entrenamientos, entonces encontrará un botón de “descargar entrenamientos” permitiendo obtener todos en un listado.
- Pruebas unitarias
 - Mensaje informativo sin entrenamientos registrados
 - Listado de entrenamientos con datos completos
 - Descarga del listado de entrenamientos

Sprint 2:

a) Gestión de niveles académicos (2 Puntos de historia)

Como administrador académico quiero crear y gestionar niveles para estructurar la malla curricular de forma jerárquica.

- Horas de trabajo: 54 horas.
- Tareas:
 - Diseñar estructura de niveles académicos.
 - Crear formulario de gestión de niveles.
 - Desarrollar tabla dinámica para la visualización de niveles.
- Criterios de aceptación:
 - Dado que administre el sistema, cuando cree un nivel nuevo entre los ya existentes, entonces puede asignarle una jerarquía en relación con los ya existentes.
 - Dado que quiera eliminar un nivel existente, cuando ingrese al gestor de niveles académicos, entonces tendrá un botón de “eliminar”, permitiendo reducir en la jerarquía los niveles existentes.
 - Dado que quiera editar los niveles existentes, cuando ingrese al gestor, entonces tendrá la opción de “editar”, permitiendo modificar su jerarquía como también su nombre.
- Pruebas unitarias
 - Creación de nivel con jerarquía asignada
 - Eliminación de nivel sin cursos asociados
 - Edición de nombre y jerarquía de nivel existente

b) Gestión de cursos y módulos (5 Puntos de historia)

Como administrador académico quiero crear y gestionar cursos con sus respectivos módulos para estructurar el contenido educativo de forma organizada.

- Horas de trabajo: 90 horas.
- Tareas:
 - Diseñar estructura jerárquica (Nivel-Curso-Módulo).
 - Crear formularios de gestión de cursos.
 - Diseñar tabla dinámica para la visualización de cursos y módulos.
 - Desarrollar asignación de instructores a módulos.
- Criterios de aceptación:
 - Dado que quiera crear un curso nuevo, cuando ingrese al gestor de cursos podrá asignar un nombre, asociarlo a un nivel y también la cantidad de alumnos capaces de ingresar a este.
 - Dado que quiera crear un módulo y haya cursos ya creados cuando ingrese al gestor asociado, entonces puede generar en base a un formulario un módulo nuevo con su curso asociado.
 - Dado que quiera modificar un curso cuando ingrese al gestor de cursos, entonces podrá eliminar o editar el contenido asociado al listado de cursos creados.

- Pruebas unitarias
 - Creación de curso con nivel y cupos válidos
 - Creación de módulo asociado a curso existente
 - Eliminación y edición de curso existente

- c) **Control de prerrequisitos (5 Puntos de historia)**

Como administrador quiero definir cursos con prerrequisitos para controlar la progresión académica y garantizar el cumplimiento de la malla curricular.
- Horas de trabajo: 108 horas.
- Tareas:
 - Diseñar tabla dinámica con la estructura de dependencias de cursos.
 - Crear formulario con la configuración de prerrequisitos.
 - Implementar validación automática de prerrequisitos.
- Criterios de aceptación:
 - Dado que quiera asociar un curso como prerrequisito de otro, cuando ingrese al gestor de cursos y edite el formulario asociado al curso, entonces puede asignar entre los cursos ya existentes pertenecientes al mismo nivel, un curso como prerrequisito.
 - Dado que quiera eliminar un curso que es un prerrequisito de otros cursos, cuando ingrese al gestor y elimine el curso en cuestión, entonces el sistema volverá a generar las posiciones restando de la cola el curso eliminado.
 - Dado que quiera modificar la posición de un curso como prerrequisito, cuando ingrese en el gestor de cursos a través del curso seleccionado, entonces puede cambiar la posición de los cursos pertenecientes a prerrequisitos.
- Pruebas unitarias
 - Asignación de curso como prerrequisito dentro del mismo nivel
 - Reordenamiento de prerrequisitos al eliminar un curso
 - Modificación de posición de prerrequisito

- d) **Reconocimiento de especialistas (3 Puntos de historia)**

Como Rector Académico de Bomberos quiero un sistema de filtrado para reconocer cuantos bomberos poseen las distintas especialidades del CBVM.
- Horas de trabajo: 108 horas.
- Tareas:
 - Crear catálogo de especialidades disponibles.
 - Diseñar interfaz de asignación de especialidades.
 - Crear visualización estadística de especialidades.
 - Implementar sistema de filtros avanzados.
- Criterios de aceptación:
 - Dado que quiera reconocer cuantos miembros del CBVM tienen la especialidad en Hazmat, cuando entre en su sistema administrativo y vaya a la sección "participantes", entonces el sistema mostrará un listado con todos los bomberos registrados, con la posibilidad de filtrarlos.
 - Dado que quiera filtrar con distintos criterios, cuando que seleccione múltiples filtros (grado + especialidad), entonces éstos se combinarán correctamente, permitiendo filtrar con mayor precisión.
 - Dado que necesite contactar a especialistas, cuando filtre según la especialidad, entonces podrá seleccionar los nombres dentro del buscador para revisar su información de contacto.
- Pruebas unitarias
 - Filtrado de bomberos por especialidad
 - Combinación de filtros por grado y especialidad
 - Acceso a información de contacto de especialista

e) Inscripción del cuerpo bomberil (5 Puntos de historia)

Como Personal Administrativo de Bomberos quiero un sistema para añadir a los distintos miembros de bomberos para que tengan acceso a sus determinadas funciones o cursos disponibles.

- Horas de trabajo: 117 horas.
- Tareas:
 - Diseñar formulario de registro de bomberos.
 - Crear sistema de validación de datos personales.
 - Desarrollar gestión de compañías y asignaciones.
 - Diseñar tabla dinámica para la inscripción de bomberos.
- Criterios de aceptación:
 - Dado que sea administrador, cuando ingrese RUN, nombre, correo, rol, además de otros datos administrativos, entonces el sistema creará un nuevo miembro activo.
 - Dado que ingrese un correo ya existente, cuando intente guardarlo, entonces obtendrá el error “Correo ya registrado”.
 - Dado que se ha creado un nuevo miembro, cuando se guarde, entonces quedará visible en la lista de usuarios con su estado “Activo”.
- Pruebas unitarias
 - Registro exitoso de nuevo miembro activo
 - Rechazo por correo electrónico duplicado o RUN
 - Viabilidad del nuevo miembro en listado con estado activo

f) Inscripción de bomberos a módulos (2 Puntos de historia)

Como administrador académico quiero inscribir bomberos en módulos específicos para gestionar su avance en la malla curricular.

- Horas de trabajo: 99 horas
 - Diseñar formulario de inscripción jerárquica.
 - Desarrollar inscripción masiva a múltiples módulos.
 - Crear historial de inscripciones con filtros por bombero.
 - Diseñar tabla dinámica para visualización de inscripciones.
- Criterios de aceptación
 - Dado que quiera inscribir múltiples bomberos a un módulo respectivo, cuando ingrese al gestor de inscripciones tendrá el botón “generar”, permitiendo entonces realizar un registro a través de un formulario con todos los alumnos posibles para ser inscritos en un módulo respectivo.
 - Dado que esté generando un registro, pero éste tiene alumnos ya inscritos en el módulo asociado, cuando intente guardar la inscripción, entonces el sistema alertará mencionando que, entre el listado de alumnos, uno específico ya está matriculado.
 - Dado que quiera matricular a un grupo de alumnos en múltiples módulos, cuando ingrese al gestor, entonces podrá seleccionar múltiples módulos como alumnos, siempre y cuando estos no posean incompatibilidades por prerrequisito.
- Pruebas unitarias
 - Generación de inscripción grupal a módulo
 - Alerta por alumno ya inscrito en el módulo
 - Validación de prerrequisitos en inscripción múltiple

g) Sistema de comunicación administrativa (3 Puntos de historia)

Como administrador quiero publicar anuncios globales para informar a instructores y alumnos sobre eventos o comunicados importantes del cuerpo de bomberos.

- Horas de trabajo: 99 horas.



- Tareas:
 - Diseñar de tabla dinámica para la gestión de anuncios.
 - Crear formulario para anuncios.
 - Desarrollar visualización de anuncios en dashboard.
- Criterios de aceptación:
 - Dado que quiera registrar un comunicado para la comunidad bomberil, cuando ingrese al gestor de anuncios, entonces podrá generar un nuevo anuncio, marcando un título, descripción y su duración.
 - Dado que quiera generar un comunicado específicamente para los instructores, cuando ingrese al formulario de creación de anuncios, entonces podrá utilizar el selector para marcar a quién irá dirigido el mensaje.
 - Dado que requiera modificar o eliminar un anuncio, cuando ingrese al gestor, entonces podrá revisar a través de una tabla dinámica todos los anuncios creados con la opción de modificar sus campos o eliminar el anuncio.
- Pruebas unitarias
 - Creación de anuncio con público objetivo definido
 - Filtrado de anuncios por público objetivo
 - Modificación y eliminación de anuncios existentes
- h) Sistema de notificaciones del instructor (3 Puntos de historia)**

Como instructor quiero enviar notificaciones a mis alumnos según el módulo o curso para mantener una comunicación constante sobre actividades, clases o recordatorios.
- Horas de trabajo: 72 horas.
- Tareas:
 - Diseñar formulario para la creación de notificaciones.
 - Implementar selección de destinatarios por módulo.
 - Implementar historial de notificaciones enviadas.
 - Crear sistema de plantillas de notificaciones.
- Criterios de aceptación
 - Dado que quiera comunicarse con sus alumnos, cuando entre al gestor de notificaciones a realizar un comunicado, entonces se mostrará un formulario para generar notificaciones al respectivo módulo.
 - Dado que requiera generar una notificación para los miembros de especialidades, cuando entre al gestor de notificaciones, entonces tendrá la opción en el formulario para seleccionar especialidades con los cursos correspondientes.
 - Dado que requiera eliminar o modificar una notificación, cuando ingrese al gestor de notificaciones, entonces podrá ver mis notificaciones creadas y modificar sus campos, como también eliminarlas.
- Pruebas unitarias
 - Creación y envío de notificaciones a módulos asignados
 - Notificación filtrada por especialidad y curso
 - Modificación y eliminación de notificación creada
- i) Estadísticas importantes (5 Puntos de historia)**

Como rector quiero visualizar gráficos que muestren progreso académico, cursos completados y distribución de especialidades, para tomar decisiones informadas.
- Horas de trabajo: 99 horas
- Tareas:
 - Diseñar dashboard para el administrador.
 - Diseñar widgets con gráficos del progreso académico.
 - Crear reportes de los niveles completados.
 - Desarrollar visualización de distribución de especialidades.



- Criterios de Aceptación
 - Dado que sea un usuario administrador, cuando ingrese al sistema, entonces podrá visualizar gráficos relacionados al conteo de miembros, como también la supervisión de sus niveles respectivo y sus especialidades.
 - Dado que necesite visualizar cuantos miembros pertenecen a una compañía en específico, cuando ingrese al sistema y seleccione el gráfico que visualiza la cantidad de miembros de cada compañía, entonces podrá visualizar cuantos miembros están inscritos en ella a través de un listado.
 - Dado que necesite comparar cuantos miembros hay en cada compañía, cuando se dirija al gráfico capaz de contabilizar los miembros del CBVM, entonces podrá utilizar vistas parciales como totales para revisar que compañías tienen más o menos miembros matriculados.
- Pruebas unitarias
 - Visualización de gráficos de miembros y niveles para administrador
 - Listado de miembros por compañía específica
 - Comparativa de miembros entre compañías

Sprint 3:

a) Gestión de alumnos inscritos (5 Puntos de historia)

Como Profesor de la Escuela Bomberil quiero un listado de los alumnos para llevar constancia de todos los bomberos que participan en mis cursos.

- Horas de trabajo: 81 horas
- Tareas:
 - Diseñar vista de alumnos por módulo.
 - Crear filtros por estado de inscripción.
 - Implementar búsqueda avanzada de alumnos.
- Criterios de aceptación
 - Dado que sea un instructor del CBVM, cuando inicie sesión, entonces podrá visualizar un dashboard capaz de proporcionar un listado resumido con los módulos con más alumnos inscritos.
 - Dado que quiera visualizar los alumnos inscritos a un módulo específico, cuando revise en su dashboard un módulo, entonces podrá seleccionar el botón de “ver inscritos” para visualizar a través de un listado los alumnos pertenecientes al módulo, y si poseen una calificación.
 - Dado que requiera visualizar todos los módulos que imparte, cuando visualice el apartado de inscripciones por módulo en dashboard, entonces podrá seleccionar el botón de ver todos los módulos para visualizar cada módulo que está impartiendo.
- Pruebas unitarias
 - Dashboard del instructor con módulos y alumnos inscritos
 - Listado de alumnos inscritos con estado de calificación
 - Listado completo de módulos impartidos por el instructor

b) Evaluación de módulos (8 Puntos de historia)

Como instructor quiero evaluar a mis alumnos en cada módulo para registrar su rendimiento académico y emitir retroalimentación.

- Horas de trabajo: 90 horas.
- Tareas:
 - Diseñar interfaz de evaluación por módulo.
 - Diseñar tabla dinámica para la asignación de notas.
 - Implementar registro de asistencia por módulo.
 - Implementar cierre y aprobación de módulos.



- Criterios de aceptación
 - Dado que quiera evaluar a los alumnos de un respectivo módulo, cuando ingrese al apartado de “mis módulos” en el sistema y seleccione un módulo, entonces podrá seleccionar el botón “evaluar módulo”, lo que me permitirá calificar a cada alumno.
 - Dado que requiera calificar a sus alumnos por su asistencia, cuando ingrese a un módulo específico en mi gestor, entonces podrá seleccionar el botón “evaluar módulo” y dar la preferencia de cómo calificarlo.
 - Dado que ya haya evaluado a todos sus alumnos y requiera cerrar el módulo, cuando ingrese al módulo evaluado, entonces podrá seleccionar el botón “cerrar módulo”, lo que permitirá a sus alumnos visualizar sus calificaciones y avisar a administración que el módulo está finalizado.
- Pruebas unitarias
 - Registro y calificación del alumno en el módulo
 - Registro de asistencia como preferencia de calificación
 - Cierre de módulo con notificación a alumnos y administración
- c) Material didáctico (5 Puntos de historia)**

Como instructor quiero subir y actualizar materiales de apoyo en los módulos que imparto para que los alumnos puedan acceder al contenido de estudio.
- Horas de trabajo: 99 horas.
- Tareas:
 - Diseñar sistema de gestión de archivos
 - Implementar carga de documentos múltiples.
 - Crear organización de material por módulo.
 - Desarrollar visualización y descarga de archivos.
- Criterios de aceptación
 - Dado que requiera subir material de estudio para sus alumnos, cuando ingrese a un módulo específico del gestor de módulos, entonces podrá visualizar el apartado “archivos del módulo”, permitiendo seleccionar un archivo y asignarle un nombre.
 - Dado que haya subido material a un módulo específico, cuando ingrese a los archivos del módulo, entonces podrá visualizar los que ha subido, como también descargarlos.
 - Dado que necesite editar un archivo que ha subido, cuando visualice el apartado de los archivos del módulo, entonces podrá modificar su contenido y su nombre, como también eliminar el archivo subido.
- Pruebas unitarias
 - Carga de archivos con nombre asignado en módulo específico
 - Visualización y descarga de archivos del módulo seleccionado
 - Edición y eliminación de material de estudio
- d) Asignación de roles (2 Puntos de historia)**

Como personal administrativo quiero asignar/quitar roles (Alumno y profesor) para controlar permisos.
- Horas de trabajo: 99 horas
- Tareas:
 - Anexar funcionalidad al gestor de inscripción de alumnos e instructores.
 - Crear sistema de permisos por rol.
 - Implementar notificaciones de cambio de permisos.



- Criterios de aceptación
 - Dado que requiera modificar el rol de un alumno, cuando ingrese al gestor del alumnado, entonces podrá modificarlo a través de “editar usuario” para visualizar sus datos y su rol pertinente.
 - Dado que deba reducir los permisos de un instructor, cuando ingrese al gestor de instructores, entonces podrá modificar su información, como también bajar su rol a alumno.
 - Dado que modifique el rol de un usuario y no guarde los cambios cuando quiera salir del gestor, entonces el sistema notificará que hay cambios sin guardar.
- Pruebas unitarias
 - Modificación de rol de alumno desde el gestor
 - Degradación de rol de instructor a alumno
 - Advertencia por cambios sin guardar al salir del gestor
- e) **Sistema de recuperación de contraseñas (8 Puntos de historia)**

Como usuario quiero poder recuperar mi contraseña o actualizar mis datos personales para mantener mi cuenta segura y actualizada.
- Horas de trabajo: 108 horas
- Tareas:
 - Desarrollar formulario de cambio de contraseña.
 - Diseñar flujo de recuperación de contraseña.
 - Anexar función de recuperación de contraseña en el gestor de alumnos e instructores.
- Criterios de aceptación
 - Dado que un alumno pierda sus credenciales, cuando ingrese al gestor del alumnado tendrá el botón de “modificar contraseña”, entonces podrá proporcionarle una contraseña nueva.
 - Dado que sea necesario reiniciar de base una contraseña, cuando ingrese a la gestión de usuarios, al apretar el botón de “modificar contraseña”, entonces podrá dar con el opcional de “cambiar contraseña al primer login”, permitiendo al usuario crear una contraseña personalizada después de su ingreso satisfactorio.
 - Dado que un instructor requiera modificar su información, cuando ingrese al gestor de instructores, entonces podrá modificar su información personal, como su usuario para iniciar sesión.
- Pruebas unitarias
 - Asignación de nueva contraseña desde gestor administrativo
 - Activación de cambio de contraseña en primer login
 - Modificación de información personal del instructor
- f) **Métricas del profesorado (5 Puntos de historia)**

Como Profesor de la Escuela Bomberil quiero un seguimiento estadístico para llevar constancia de la asistencia de mis alumnos en los respectivos módulos.
- Horas de trabajo: 99 horas.
- Tareas:
 - Diseñar dashboard del instructor.
 - Implementar visualización de asistencia por módulo.
 - Implementar comparativas entre módulos.
- Criterios de aceptación
 - Dado que haya calificado la asistencia de sus alumnos, al visualizar en el dashboard las inscripciones por módulo en el apartado de “ver inscritos”, podrá visualizar a los alumnos y su progreso de la asistencia al módulo.
 - Dado que no haya calificado toda la asistencia de sus alumnos al visualizar el gráfico relacionado a la asistencia del curso, el conteo del total de alumnos aparecerá en rojo, significando que hay evaluaciones pendientes.



- Dado que se requiera visualizar el total de módulos con su respectiva asistencia al seleccionar ver todos los módulos, entonces será posible visualizar el total de módulos impartidos con su respectiva asistencia porcentual.
- Pruebas unitarias
 - Visualización de progreso de asistencia por módulo
 - Indicador de evaluaciones de asistencia pendientes
 - Listado de módulos con asistencia porcentual
- g) Asistente chatbot para reportes de gestión administrativa (13 Puntos de historia)**

Como usuario administrador del sistema quiero interactuar con un chatbot que me permita encontrar información sobre alumnos, módulos o instructores para agilizar mi búsqueda dentro del sistema.
- Horas de trabajo: 117 horas
- Tareas:
 - Configurar integración con API de inteligencia artificial.
 - Diseñar interfaz de chat interactivo.
 - Implementar contexto bomberil en el asistente.
 - Crear sistema de consultas personalizadas.
- Criterios de aceptación
 - Dado que requiera realizar una consulta al chatbot “asistente TechBot” sobre la cantidad de especialistas de un área en específico, cuando genere la consulta, entonces se generará un listado con los alumnos correspondientes con la capacidad de descargarlo como un archivo de texto.
 - Dado que realice una consulta de datos inexistentes dentro del sistema, cuando realice la consulta sobre este tipo de dato, entonces el asistente notificará que éstos no existen.
 - Dado que realice muchas consultas dentro de un día, alcanzando las 20 solicitudes, cuando llegue a la veintiunava solicitud, entonces el sistema notificará que ha superado el límite de tokens diarios.
- Pruebas unitarias
 - Consulta de especialistas por área con listado descargable
 - Respuesta ante consulta de datos inexistentes
 - Bloqueo por límite diario de consultas alcanzado

Las historias de usuario *Autenticación de usuario*, *Dashboard* y *perfil de usuario* e *Certificaciones de la malla curricular* constan de la base principal del sistema, buscando suplir la necesidad de nuestro cliente, la cual consta de la incapacidad de generar un seguimiento a tiempo real de las certificaciones de los miembros del CBVM, como también un sistema virtualizado para acceder cómodamente a la documentación correspondiente.

Las historias de usuario *Gestión de niveles académicos*, *Gestión de cursos y módulos*, *Inscripción del cuerpo bomberil* y *Inscripción de bomberos a módulos* constan de las bases para diseñar un sistema académico, capaz de emular la malla académica, como también inscribir a tanto a los instructores como al alumnado en los distintos módulos a dictar en los semestres académicos, lo que permitirá al cuerpo bomberil llevar un seguimiento de los módulos a impartir como también proporcionar continuidad en el transcurso del año académico.

Las historias de usuario *Gestión de alumnos inscritos*, *Evaluación de módulos*, permiten a los bomberos proporcionar un seguimiento virtual y una continuidad a tiempo real de estado de los módulos realizados, pudiendo enterarse a tiempo real el avance de su malla curricular; *Sistema de recuperación de contraseñas*, *Asistente chatbot para reportes de gestión administrativa* permiten a los administradores mantener un mayor control en la gestión de usuarios como también un reconocimiento

rápido y efectivo de los distintos miembros del CBVM, siendo capaz de reconocer a los distintos especialistas dentro del sistema en cuestión de segundos.

4.4 Planificación temporal del proyecto – Roadmap del sistema

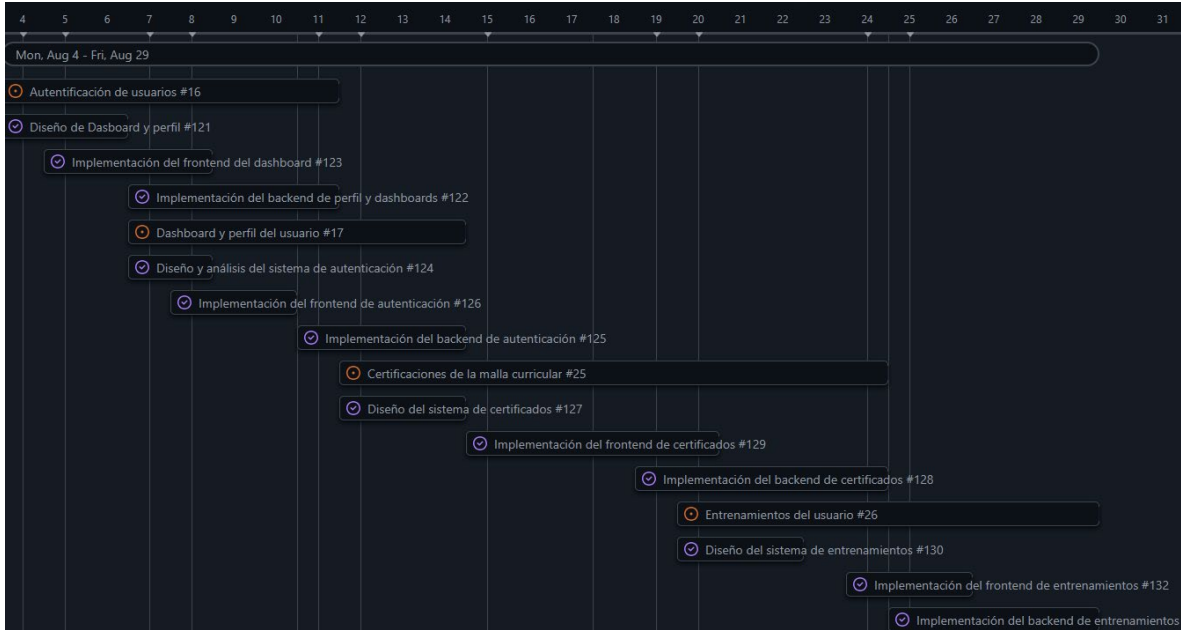


Ilustración 2 – Roadmap con funcionalidades y tareas de la iteración 1

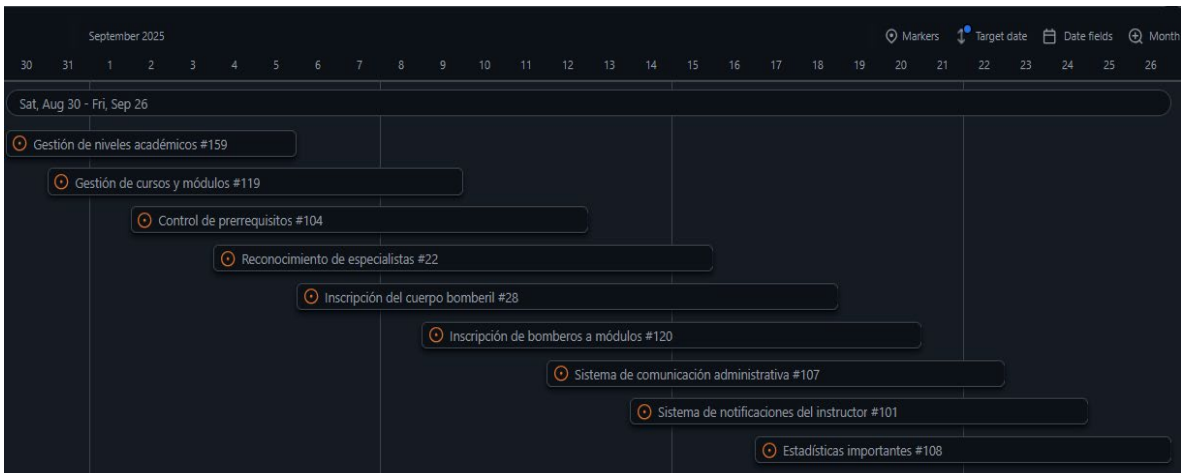


Ilustración 3 – Roadmap con funcionalidades de la iteración 2



Ilustración 4 – Roadmap de las funcionalidades de la iteración 3

5 Ejecución de reuniones ágiles

5.1 Descripción de reuniones diarias, de planificación, revisiones y retrospectivas

Bajo la metodología Scrum se realizan diferentes eventos, los cuales permiten desarrollar en las distintas iteraciones con objetivos estratégicos, tácticos y operativos, lo que permite al equipo coordinar un desarrollo incremental en concordancia con la satisfacción y las necesidades del cliente como también la entrega de valor correspondiente.

Para planificar en relación con la fijación de objetivos, el equipo decide adaptar los eventos principales de la metodología, esto según las necesidades en cada iteración; dejando un mínimo de 3 Daily Scrum semanales, buscando solventar las distintas necesidades transcurridas en el desarrollo de software como también la fijación de objetivos operativos en correlación al avance en las funcionalidades del sistema.

De esta manera, los eventos de planificación del sprint se realizan al iniciar la iteración y, la revisión del sprint, como también la retrospectiva del sprint, son reuniones realizadas al finalizar cada iteración, las cuales estarán integradas para realizar el entregable de la bitácora a excepción de la planificación del sprint; el perfeccionamiento de nuestras historias de usuario será realizado por semana para evaluar el desempeño del equipo y reestimar el progreso del desarrollo de software, permitiendo a nuestro equipo priorizar y modificar las tareas introducidas en el *Product Backlog* de la iteración a realizar.

La revisión del sprint se realiza mediante canales digitales con el patrocinador, permitiendo así, coordinar independiente de los contratamientos del equipo, participar en la recolección de requisitos, como también resolver dudas en cuanto a la lógica subyacente a funcionalidades desarrolladas, cómo la lógica de negocio, en este caso la lógica académica que requiere el CBVM para estructurar el sistema en relación al desarrollo de gestores administrativos como también el correcto funcionamiento de la malla académica.



5.2 Importancia de la comunicación y colaboración en el equipo

La colaboración y la comunicación son esenciales en el desarrollo de software, la gestión de proyectos y la entrega de productos para las metodologías ágiles no son la excepción, dentro de los aspectos importantes a destacar se pueden encontrar distintos factores que son el motor para alcanzar el éxito en un proyecto.

- Comunicación mejorada:
 - La comunicación abierta mediante reuniones diarias, una planificación iterativa y asistido de herramientas colaborativas capaz de garantizar que se mantengan alineados los objetivos del proyecto, reducen malentendidos y retrasos.
- Resolución de problemas más rápida
 - Cuando los miembros del equipo trabajan juntos, pueden identificar y resolver problemas rápidamente, la resolución colaborativa de obstáculos aprovecha la experiencia colectiva del grupo, impulsando soluciones innovadoras.
- Mayor productividad
 - La colaboración minimiza los malentendidos y fomenta la colaboración entre equipos para alcanzar objetivos comunes. Esta alineación genera mayor eficiencia y un flujo de trabajo optimizado, ya que las tareas se comparten y se priorizan eficazmente.
- Aprendizaje y mejora continua
 - La metodología ágil se basa en retrospectivas y ciclos de retroalimentación, que prosperan en entornos colaborativos, los equipos aprenden de los éxitos y los fracasos, mejorando continuamente sus procesos.
- Moral de equipo más fuerte
 - La colaboración fomenta el sentido de pertenencia y la responsabilidad compartida entre los miembros del equipo. Esto mejora la moral y crea una cultura laboral positiva.
- Técnicas de preguntas eficientes y escucha activa
 - La escucha activa permite formular preguntas aclaratorias, buscando comprender la perspectiva de los demás; las técnicas de interrogatorio eficaces, como las preguntas abiertas, ayudan a obtener información significativa y fomentan respuestas reflexivas.
- Empatía e inteligencia emocional en la comunicación
 - Estas habilidades son esenciales para una comunicación eficaz, comprender, considerar las emociones y tomar en cuenta las perspectivas de los demás crea un entorno de apoyo y colaboración.

6 Gestión de cambios y mejora continua

6.1 Adaptación a cambios en los requerimientos

El proyecto es abordado por la academia de bomberos de Viña del Mar, condicionado al desarrollo iterativo de software, se fueron conformando requisitos y modificaciones necesarias de abordar en la mayoría de los casos, como también limitar al alcance debido a los plazos preestablecidos, estos cambios implican:

- Refinar las Historias de usuario
 - Conforme al avance iterativo en el desarrollo de software, se toma en cuenta las necesidades del cliente a la hora de proporcionar entregables, replanteando las historias usuarias, mejorándolas, igualmente modificando la lógica del requisito a desarrollar.
- Redefinición del alcance
 - En relación con la reestructuración de historias de usuario, y la recepción de nuevos requisitos, se toma en cuenta las funcionalidades a priorizar versus requisitos planteados en instancias iniciales.
- Gestión de riesgos
 - Identificar riesgos técnicos y organizativos, tomando un plan preventivo en caso de sufrir un escenario dentro del margen analizado.
- Backlog dinámico
 - Incorporando los comentarios de los interesados del proyecto, se estima nuevamente las historias de usuario como las tareas de desarrollar en la iteración siguiente.

6.2 Gestión de mejoras basadas en feedback

Entre los valores del manifiesto ágil están la entrega de valor y la mejora continua, estos puntos esenciales son posibles de ser realizados con una comunicación continua con los *stakeholders* del proyecto, es por esto, que se instancian reuniones retrospectivas para obtener un *feedback iterativo* lo que permite perfeccionar el producto y los procesos del equipo.

Entre las mejoras basadas en el feedback generado por los patrocinadores es posible encontrar:

- Modificaciones al alcance del proyecto
 - Debido a funcionalidades nuevas a integrar se discutió con el rector académico las funcionalidades prioritarias en el sistema, de las cuales, las que no fuesen posibles de ingresar por limitaciones de tiempo, serían integradas en un nuevo MVP.
- Modificaciones a la lógica de las funcionalidades
 - En base a las reuniones retrospectivas, se logra comprender la lógica requerida por el cliente, lo que implica modificar funcionalidades acordes a las dolencias actuales del patrocinador.
- Reestimar la priorización de funcionalidades a desarrollar
 - Al adjuntar nuevos requisitos y cambiar la lógica de las funcionalidades ya desarrolladas, se considera nuevamente estimar y priorizar el ciclo de desarrollo de software.

6.3 Tabla de las Historias de usuario a Refinar

Historia usuaria original	Motivo	Refinamiento	Funcionalidades incorporadas
Como personal administrativo de bomberos quiero poder registrar a los bomberos que no pudieron anexarse en el sistema pre-2021 para llevar constancia de los cursos que realizaron y el grado en el cual están actualmente.	La (HU) consta de demasiados requisitos, en donde está incluido el ingreso de usuarios, la destinación de cursos, el registro del nivel actual del bombero, como también los módulos realizados o pendientes.	-Se creo una (HU) para matricular alumnos e instructores. -Se crean gestores para llevar un manejo jerárquico de los niveles, cursos y módulos. -Se implementan gestores para inscribir alumnos en los módulos a dictaminar.	-Inscripción del cuerpo bomberil. -Gestión de niveles académicos. -Gestión de cursos y módulos. -Inscripción del cuerpo bomberil.
Como instructor quiero marcar asistencia de mis alumnos en cada sesión para llevar un control académico y operativo del curso.	La (HU) consta de demasiadas funcionalidades como registrar las sesiones que el instructor realice sus clases, como también un sistema que pudiese dividir los bloques horarios para no generar choques entre instructores, son demasiadas horas de trabajo para realizar dichas funcionalidades.	-Se creó un sistema para visualizar los alumnos inscritos a cada módulo - Se creo un sistema para marcar la asistencia, como también registrar una calificación al evaluar el módulo.	-Gestión de alumnos inscritos. -Evaluación de módulos.
Como bombero quiero descargar los certificados de los cursos ya realizados para tener una constancia del grado actual en el que estoy y tener constancia de la especialidad ya realizada.	La (HU), consta de demasiados requisitos para ser abordable en una sola iteración, por lo que se opta por dividir el trabajo en la descarga de certificados, entrenamientos y en las especialidades realizadas.	-Se implementó un gestor para que los administradores del sistema adjunten los certificados del alumnado. -Se generó una vista para que los bomberos descarguen los certificados emitidos. -Se implementó un sistema para reconocer las especialidades.	-Certificados de la malla curricular. -Entrenamientos del usuario. -Reconocimiento de especialistas.

Tabla 1 - Historias de usuario a refinar

6.4 Tabla de Gestión del alcance

Iteración	Cambio	Horas de trabajo estimadas	Resolución del cambio	Motivo
Iteración 2	La brigada juvenil, alumnos menores de edad deben presentar un formulario adjuntado a su tutor legal, validando su ingreso.	70 horas	Rechazado	Se opta por no permitir el ingreso a menores de edad, por lo que los miembros de la brigada juvenil no pueden ingresar al sistema.
iteración 2	El login debe poseer un "recuérdame" para mantener las sesiones.	50 horas	Rechazado	Proporcionar un sistema de recuérdame da el paso a generar vulnerabilidades, por lo que es preferible extender la duración del token.
Iteración 2	El login del sistema debe estar asociado por Microsoft.	90 horas	Rechazado	Reestructurar todo el sistema de inicio de sesión no se asocia con los objetivos de la iteración y nos llevaría a eliminar toda una funcionalidad.
Iteración 2	La información relacionada a la ANB debe eliminarse debido a que el CBVM no posee acceso a la base de datos.	20 horas	Aceptado	Reestructurar el perfil del usuario permite acoplar nuevas funcionalidades, sin entregar una vista tan relacionada a solo proporcionar datos del usuario.
Iteración 2	Deben existir métodos para anexar usuarios sin la necesidad pasar por los desarrolladores de software.	100 horas	Aceptado	Debido a los datos sensibles de los usuarios surge la necesidad de delegar la capacidad de registrar usuarios reales a los bomberos, por lo que será necesario estructurar gestores para que ellos registren a sus usuarios.
Iteración 3	El manejo de asistencia por clase debe ser modificado a una evaluación final.	70 horas	Aceptado	En base a la resolución con bomberos explicando los motivos del porque no podía ser manejado por clases la asistencia, se llega al consenso que la asistencia pueda ser manejada al final de cada módulo.

Tabla 2 - Gestión del alcance



Iteración	Cambio	Horas de trabajo estimadas	Resolución del cambio	Motivo
Iteración 3	El sistema debe generar reportes descargables por el nivel de los usuarios, especialistas, distribuidos por compañía.	120 horas	Rechazado	El sistema chatbot permite reconocer a los distintos tipos de usuarios distribuidos por compañías, clasificándolos por nivel y por especialidad.
Iteración 3	El calendario debe marcar las notificaciones, los eventos y los periodos donde el instructor ingresa la evaluación del módulo.	90 horas	Rechazado	Las funcionalidades relacionadas al calendario fueron removidas debido a las horas extras de trabajo que significan anexar distintas funcionalidades para que el calendario funcione.
Iteración 3	El instructor no debe poseer la capacidad de crear cursos ni módulos.	30 horas	Aceptado	El remover la capacidad del instructor implicó generar limitación de accesos, una funcionalidad necesaria para la navegación por roles.
Iteración 3	El control de la asistencia debe ser manejado a nivel de módulos no a nivel de cursos.	75 horas	Aceptado	Registrar asistencia en módulos permite anexar la funcionalidad en la capacidad de registrar calificaciones.

Tabla 3 - Gestión del alcance

6.5 Tabla gestión de riesgos

Riesgo identificado	Impacto	Probabilidad	Acciones preventivas	Acciones correctivas
No completar todas las tareas en la iteración.	Alto	Media	Limitar las funcionalidades del requisito, proporcionando un diseño simple y sin la necesidad de una lógica compleja.	Aceptar una deuda técnica para la iteración siguiente, tomando una carga mayor de trabajo.
No lograr concretar una reunión con el patrocinador.	Alto	Alta	Seguir la línea de trabajo establecida en reuniones anteriores, manteniendo el foco en las principales dolencias del cliente.	Enviar una demo de las funcionalidades realizadas por el equipo, buscando una retrospectiva.
No coordinar con el equipo desarrollador reuniones operativas.	Medio	Media	Dejar un canal de comunicación alternativo, para que el equipo pueda dar a conocer sus limitaciones del momento como también las dudas que van surgiendo.	Reforzar la importancia de mantener una comunicación en voz, lo que permite una mayor coordinación en la resolución de entregables e impedimentos.
Realizar cambios en Github que alteren las funcionalidades del sistema.	Alto	Baja	Realizar reuniones para el manejo correcto de los repositorios de código como también dar a conocer comandos seguros para no causar conflictos en los espacios de trabajo.	Realizar un respaldo sin los cambios implementados por el desarrollador.
Un desarrollador no logra completar las tareas planteadas para la semana.	Alto	Baja	Delegar tareas en base a las estimaciones entre grupos de trabajo, entre los cuales coordinarán las tareas a repartir.	Un desarrollador de encargado de otra tarea suplirá las tareas del otro desarrollador y en caso de no ser cumplida en el plazo, será llevada como deuda técnica para la siguiente iteración.
Fallas en la integración de versiones a la hora de realizar funcionalidades.	Medio	Baja	Realizar espacios de trabajo apartados, en donde existen 2 ramas principales de trabajo, una llamada main en donde solo ingresan funcionalidades estables y develop un espacio de trabajo para ingresar funcionalidades y resolver problemas en la integración.	Buscar el origen de la falla y generar una nueva subrama con la corrección, para posteriormente ser integrada en develop.
Fallas técnicas a la hora de subir el sistema a la red.	Medio	Media	Trabajar exhaustivamente en la estabilidad de funcionalidades en main, entendiendo la lógica subyacente de cada funcionalidad.	Realizar cambios relacionados a hotfix, corrigiendo a tiempo real los errores detectados.

Tabla 4 - Gestión de Riesgos

7 Evaluación del desempeño ágil

7.1 Métricas de evaluación

Para analizar el desempeño del proyecto se usarán métricas para cada iteración, analizando cuantas funcionalidades fueron realizadas, que tan lejos estamos de realizar un MVP y cuál es la deuda técnica que estamos llevando a la siguiente iteración, carga de trabajo adicional, acarreado funcionalidades de la iteración anterior.

Basado en esto, se toma en cuenta a considerar una historia completada al superar un margen del (90%), entendiendo que, la historia al estar cerca de finalizar la iteración está en un margen de validación, o sea que, aún está recibiendo pruebas unitarias, para verificar que el funcionamiento de esta no vaya afectar el ejecución del sistema en futuras iteraciones; Al superar este margen de validación, la historia se considera completada (100%) y por lo tanto, el equipo puede priorizar las distintas funcionalidades a testear de la iteración; en caso que las iteraciones superen el margen de los 3 sprints, tenemos un de tiempo adicional hasta la feria de software (13 de noviembre), en el cual el equipo se centra en subir la aplicación y terminar las pruebas necesarias para completar el total de las funcionalidades.

Métricas por analizar:

- Historias de usuario completadas sobre el 90% completado
- Porcentaje del MVP
- Porcentaje de la funcionalidad Realizada
- Deuda técnica de la iteración
- Deuda técnica de la iteración anterior

Sprint 1

Historias de usuario completadas sobre el 90%: 3 / 4

1. Autenticación de usuarios (100% completado)
 2. Dashboard y perfil del usuario (90% completado)
 3. Certificaciones de la malla curricular (70% completado)
 4. Entrenamientos del usuario (100% completado)
- Deuda técnica de la iteración: 2 Historias de usuario, 1 funcionalidad en validación
 - Porcentaje del MVP: 10% completado

Sprint 2

Historias de usuario completadas sobre el 90% completado: 7 / 9

1. Gestión de niveles académicos (100% completado)
2. Gestión de cursos y módulos (90% completado)
3. Control de prerrequisitos (80% completado)
4. Reconocimiento de especialistas (95% completado)
5. Inscripción del cuerpo bomberil (100% completado)
6. Inscripción de bomberos a módulos (95% completado)
7. Sistema de comunicación administrativa (100% completado)
8. Sistema de notificaciones del instructor (100% completado)
9. Estadísticas importantes (85% completado)



- Deuda técnica de la iteración: 2 Historias de usuario, 3 funcionalidades en validación
- Porcentaje del MVP: 55% completado
- Deuda técnica de la iteración anterior: 1 funcionalidad en validación (Funcionalidad 3)

Sprint 3

Historias de usuario completadas sobre el 90%: 7 / 7

1. Gestión de alumnos inscritos (100%)
 2. Evaluación de módulos (90%)
 3. Material didáctico (95%)
 4. Asignación de roles (100%)
 5. Sistema de recuperación de contraseñas (100%)
 6. Métricas del profesorado (90%)
 7. Asistente chatbot para reportes de gestión administrativa (90%)
- Deuda técnica de la iteración: 4 funcionalidades en validación
 - Porcentaje del MVP: 90% completado
 - Deuda técnica de la iteración anterior: 1 funcionalidad en validación (Funcionalidad 3)

7.2 Análisis de resultados

Debido a distintas complicaciones dentro del desarrollo del proyecto, se fue generando una deuda técnica que fue incrementando conforme al avance las iteraciones, problema que también ocurrió bastante fue el impedimento de lograr probar todas las funcionalidades antes del término de plazo de los sprint, lo que nos llevó a nuestra última iteración a depender completamente del plazo hasta la feria de software para lograr subir el programa con un sistema de nube, como también poder validar el funcionamiento de todos los requisitos que desarrollamos, lo que nos dejó a 10% de completar el total del alcance planteado, este siendo modificado en su mayor parte en la iteración 2, lo que hizo que la carga de trabajo aumente exponencialmente.

A pesar de las implicancias encontradas conforme avanzamos en el desarrollo de software, logramos mostrar un producto completado alrededor del 100%, esto siendo un estimado debido a la hora de subir el programa se encontraron problemas menores, que solo afectan a nivel visual al sistema, de igual manera logramos satisfacer las necesidades de nuestros bomberos y actualmente tienen el acceso a testear el funcionamiento del sistema para posteriormente entregar el sistema completamente a su disposición.

Como parte de las observaciones del público en la feria de software (FESW), logramos llegar de distintas implicancias para desarrollar posteriormente un nuevo MVP, como también bastantes comentarios positivos en base al sistema desarrollado, abriendo oportunidades a resolver con nuestra propuesta, problemáticas relacionadas en distintos rubros, lo que nos motiva a continuar desarrollando este proyecto en busca de auxiliar a distintas comunidades.

8 Desafíos y soluciones

En el transcurso de este proyecto, el equipo se enfrentó a distintos desafíos, hubo sesiones extensas para lograr llegar a una resolución que satisficiera a todos los miembros del equipo, como también hubo decisiones que en prácticamente minutos cambiaron el rumbo completo de nuestro proyecto, las metodologías ágiles permitieron al equipo organizarse para resolver todos los obstáculos que se detectaron.

Desafíos del proyecto

- Cambio de patrocinador
 - Debido a dificultades fuera de nuestro control, nos vimos obligados a cambiar de proyecto en dos ocasiones, lo que dificultó constantemente el avance continuo de nuestro proyecto de software.
- Adaptarse a nuevas tecnologías
 - Al buscar un sistema sostenible en el tiempo, nos dedicamos a investigar tecnologías, para que todos podamos trabajar en el proyecto de manera cómoda, el buscar equilibrar el desarrollo de una solución sostenible ante tecnologías conocidas, fue una discusión constante en la planificación del proyecto.
- Comunicación no iterativa con el patrocinador
 - El patrocinador ante su limitado tiempo siendo bombero y cumpliendo con sus responsabilidades como profesional, se hacía imposible mantener una comunicación constante.
- Tiempos limitados
 - Al comenzar desde el segundo semestre académico con la solución a desarrollar, nuestros tiempos para planificar y diseñar estrategias para bordar el sistema se vieron extremadamente limitados.
- Deuda técnica alta
 - Al abordar muchas funcionalidades para entregar una solución mínima para los patrocinadores no logramos completar en su totalidad todos los requisitos planificados en cada iteración.
- Exceso de requisitos
 - Para entregar el sistema planteado, se requiere una gran cantidad de requisitos, considerando la escalabilidad del sistema, como también la forma jerárquica en la que opera la academia de bomberos Viña del Mar.
- Virtualizar un sistema completo
 - El sistema completo es manejado a nivel físico, donde hay indicios leves de virtualización, niveles insuficientes para generar una solución relacionada a otro sistema manejado por el CBVM.
- Distribución de roles
 - El considerar estrictamente nuestros roles iniciales en el proyecto, limita nuestra capacidad de desarrollo y de generar soluciones ante las problemáticas encontradas en el desarrollo inicial del proyecto.
- Información limitada
 - Debido a los datos sensibles del CBVM, no pudimos tener los documentos asociados a la estructura exacta de como manejan su sistema a nivel presencial.
- Dificultades para coordinar reuniones
 - Debido a la carga académica de los distintos miembros del equipo, surgen dificultades a la hora de realizar reuniones asociadas a la metodología scrum.



Soluciones a los obstáculos

- Al cambiar de patrocinador al finalizar el primer semestre, buscamos métodos para comunicarnos con nuestro nuevo cliente lo antes posible, buscando tener un canal de comunicación alternativo para lograr comunicar avances y obtener feedback.
- Buscamos lenguajes en donde todos nos pudiéramos adaptar, llegando a un consenso entre todos los miembros del equipo.
- Al no poder comunicarnos iterativamente con el patrocinador, generamos demos para enviar al patrocinador, adjuntando las funcionalidades realizadas en cada sprint, de esta manera al no poder concretar reuniones con los interesados, de igual manera se puede mantener informado de los avances de los desarrolladores.
- Nuestros tiempos fueron altamente limitados, planificando, diseñando, desarrollando y testeando las distintas funcionalidades a implementar, por lo que al entender que necesitamos más desarrolladores, se fusiona el rol de Scrum Master y Product Owner para tener un rol adicional para desarrollar las distintas tareas asociadas a los requisitos.
- Lo que se relaciona con cambios en el alcance del proyecto y la refactorización de historias usuarias, el equipo cambió su metodología para abordar las distintas funcionalidades del sistema, donde un equipo se encarga de planificar y diseñar, mientras otros desarrolladores se encargan de desarrollar el front y un equipo se encarga del backend, buscando reducir la deuda técnica al finalizar los sprints.
- Poseer muchos requisitos en nuestro sistema nos lleva a poseer una deuda técnica alta, tiempos limitados, por lo que el equipo optó por limitar el alcance, eliminando requisitos que no presentan un grado alto de prioridad.
- Al no tener incidencias en un sistema virtual, se nos dificulta como abordar distintas funcionalidades del sistema, por lo que nos centramos en investigar y generar bocetos en base a los resultados de nuestras investigaciones.
- Al limitarse inicialmente a los roles seleccionados para desarrollar un proyecto de título, no logramos desarrollar las tareas relacionadas a la iteración, por lo que ajustamos los roles enfocándonos en el área que mejor nos desempeñamos.
- Debido a la limitada información nos orientamos por sistemas ya conocidos y sugeridos por el CBVM para modelar nuestro sistema, buscando añadir opciones innovadoras que logren satisfacer las necesidades del cliente.
- Optamos por realizar mínimamente 3, y en casos extremos solo 2 a la semana, destinando los objetivos de las reuniones diarias a ser comunicadas por medios alternativos.

9 Estimación de costos

El plan de costos del proyecto está pensado para los costos asociados al margen de estructuración del sistema proyectado al primer año, por lo que están involucrados los costos de compra única, como es la máquina capaz de alojar los servidores del sistema, las horas de trabajo involucradas de los desarrolladores y las horas estimadas de la gestión asociada al proyecto; los otros gastos están pensados para la mantención del sistema informático proyectados a un año de servicio, como son los gastos del nombre de dominio, el almacenamiento en nube, la base de datos, la api capaz de alimentar el chatbot y la electricidad para mantener el servidor.

Categoría	Elemento	Descripción	Costo unitario (CLP)	Cantidad	Costo total (CLP)
Infraestructura y Hosting	Servidor físico	Servidor básico para alojar el servidor frontend y backend.	600.000	1	600.000
	Azure Blob Storage (50Gb)	Servicio de nube para el almacenamiento de certificados y material didáctico.	24.000	1	24.000
	Nombre de dominio	URL del dominio, establecido en Chile.	12.000	1	12.000
	Base de datos	Estructura Azure con PostgreSQL.	58.000	12	696.000
Servicios externos y Apis	Google Gemini API	Chatbot inteligente con 500.000 tokens al día.	20.000	12	240.000
Recursos humanos	Scrum Master/Product owner (1)	Costo por horas de trabajo estimadas.	7.000	335	2.345.000
	Developers (4)	Costo por horas de trabajo estimadas.	7.000	335	9.380.000
Otros costos	Electricidad y conectividad	Costos asociados a la mantención del servidor.	180.000	1	180.000

Total estimado: 13.477.000 Pesos

Tabla 5 - Costos a asociados a la estructuración del sistema

Posterior al primer año, en donde se desarrolla la solución de software y se establecen costos únicos, como también costos asociados a la mantención del sistema, se establecen costos para proporcionar un sistema capaz de funcionar el 99,99% del tiempo, integrando como categoría el mantenimiento y soporte del sistema, categoría esencial para estructurar un servicio capaz de evitar fallas críticas y generar soluciones eficientes ante fallas, donde también es incluido las copias de seguridad en caso de fallos que puedan inhabilitar el servicio temporalmente.

Categoría	Elemento	Descripción	Costo unitario (CLP)	Cantidad	Costo total (CLP)
Mantenimiento y soporte	Mantenimiento de bugs críticos	Corrección de bugs críticos.	500.000	1	500.000
	Actualizaciones de seguridad	Parches y actualizaciones.	300.000	1	300.000
	Backup y recuperación	Copias de seguridad automatizadas.	100.000	1	100.000
Infraestructura y Hosting	Azure Blob Storage (50Gb)	Servicio de nube para el almacenamiento de certificados y material didáctico.	24.000	1	24.000
	Nombre de dominio	Url del dominio, establecido en Chile.	12.000	1	12.000
	Base de datos	Estructura Azure con PostgreSQL.	58.000	12	696.000
Servicios externos y Apis	Google Gemini API	Chatbot inteligente con 500.000 tokens al día.	20.000	12	240.000
Otros costos	Electricidad y conectividad	Costos asociados a la mantención del servidor.	180.000	1	180.000

Total estimado: 2.052.000 pesos

Tabla 6 - Costos asociados a la mantención del sistema posterior al primer año de uso

10 Conclusiones

En base al desarrollo de software bajo las metodologías ágiles, nos fue posible completar alrededor del 100% de nuestro MVP, cumpliendo el objetivo general de proporcionar una plataforma de gestión académica capaz de generar certificaciones para el CBVM, iterando con modificaciones al alcance, refactorización de requisitos y un enfoque centrado en la entrega de valor hacia el cliente; logrando a través de tres iteraciones en un lapso de 3 meses y 1673 horas de trabajo por equipo de 5 personas, priorizando el sistema de obtención de certificados como los gestores de académicos.



Objetivos específicos:

- I. Organizar los distintos requisitos del cliente, definiendo objetivos claros, realistas y abordables según las capacidades del equipo desarrollador.

La mezcla de funciones de Scrum Master y Product Owner fue fundamental para la toma de decisiones, tomando en cuenta requisitos complejos extensos y muchas veces necesario de más de 2 desarrolladores para abordar las funcionalidades, la priorización y la gestión del alcance permitió aliviar la carga en relación al desarrollo de software del equipo; por otro lado, hubo demasiada estimación en las capacidades del equipo, que en casos adversos hubiera llevado a abandonar funcionalidades completas por los plazos limitados para el nivel de requisitos.

- II. Dirigir, planificar, organizar y controlar el desarrollo de software del proyecto bajo un marco ágil, mediante herramientas asociadas a la gestión de proyecto.

Para la dirección, organización y control del desarrollo de software, se implementó como principal plataforma GitHub, en su apartado de gestión de proyectos, el manejo de una plataforma capaz de unificar el desarrollo de requisitos como la dirección del sistema, permitió contemplar un control en los plazos, las funcionalidades a desarrollar, como decidir que camino tomar ante adversidades como fue la generación de deuda técnica, realizar los eventos de la metodología scrum permitió al equipo una mayor organización para abordar las tareas a desarrollar en cada iteración como planificar la priorización de las Historias Usuarías.

- III. Realizar reuniones con el equipo desarrollador para el cumplimiento de los estándares establecidos por el grupo proyecto, con la combinación de roles de Scrum Master y Product Owner.

Generar rutinas de comunicación fue una dificultad constante, debido a las diferencias entre la disponibilidad del equipo desarrollador, buscando soluciones a la problemática constante, se optó por adaptar los principales eventos de la metodología ágil, lo que nos permitió cumplir mayoritariamente con los estándares pensados inicialmente en nuestro proyecto, pensando principalmente en nuestro objetivo común, en dar una solución viable y perdurable en el tiempo.

- IV. Validar la elaboración de diseños y vistas para la implementación de interfaces acordes al alcance del proyecto.

La supervisión en la validación de diseños nos permitió al equipo no generar tareas adicionales, lo que podría involucrar en un aumento en los plazos planificados, por lo que, limitar las vistas a la funcionalidad a desarrollar sin considerar futuras áreas de desarrollo permitió al equipo centrarse en los requisitos a desarrollar, coordinando posteriormente las pruebas unitarias.

- V. Comunicar a los interesados el avance del proyecto mediante la elaboración de reuniones y *demos*.

La comunicación fue una problemática constante con bomberos, al poseer tiempos extremadamente limitados no fue posible conversar cara a cara con nuestros clientes, por lo que optamos inicialmente a realizar videoconferencias a modo de solucionar esta problemática y acordando con el CBVM manejar un medio de comunicación alternativo, capaz de coordinar las videoconferencias, como también enviar versiones demostrativas del programa (*demos*) para generar retrospectivas de manera constante.



- VI. Mantener la cantidad de funcionalidades acorde a la capacidad de desarrollo de los programadores.

Controlar el alcance fue un problema constante en el desarrollo del proyecto, fue complicado coordinar con bomberos las funcionalidades del sistema, ya que habían diferencias entre las capacidades del equipo, asociado a nuestro tiempo limitado, relacionado con el cambio de nuestro proyecto a finales del semestre anterior, la solución que permitió abordar las diferencias del alcance fue destinar un cuarto desarrollador a realizar las tareas asociadas a las funcionalidades, permitiendo distribuir de mejor manera los entregables y llegar a un consenso de esta manera al poseer una mayor capacidad de desarrollo.

Recomendaciones para futuras iteraciones:

- Planificar pruebas unitarias y contemplar entornos para realizar pruebas usuarias como también pruebas de integración, para contemplar en primera instancia el diseño de interfaces usuarias sea totalmente legible para el cliente; por otro lado, para contemplar errores de manera temprana en el funcionamiento de interfaces asociadas con otras funcionalidades.
- Limitar el alcance del proyecto con el cliente desde etapas tempranas, buscando reducir modificaciones al alcance del proyecto como también un aumento exponencial en la carga de trabajo.
- Mejorar la comunicación ante impedimentos relacionados al abordar tareas relacionadas al desarrollo de software, y abrir la posibilidad a pedir recomendaciones entre los compañeros desarrolladores.
- Generar acuerdos previos con los interesados del proyecto para acceder a información sensible relacionada al sistema a estructurar.
- Planificar medios de comunicación estáticos desde la primera comunicación con el cliente, para reducir las diferencias desarrolladas en posteriores etapas de desarrollo, como también concretar una participación ante la retrospectiva de entregables.

Impacto y aprendizajes

Pese a todos los impedimentos en el transcurso del proyecto, TechResQ es capaz de adaptarse a las metodologías ágiles, proporcionando una solución precisa a los problemas principales del CBVM y de la academia de bomberos Viña del Mar, proporcionando un programa capaz de virtualizar un sistema totalmente manejado a nivel presencial, como también competir con desarrolladoras de alto nivel. Los aprendizajes clave obtenidos en el transcurso del proyecto se engloban.

- La capacidad de toma de decisiones ante situaciones adversas como también críticas.
- Tomar acciones preventivas ante escenarios desfavorables. adaptándose a las circunstancias.
- Limitar el alcance del proyecto y contemplar temas legales desde instancias iniciales.
- La importancia de desarrollar una planificación estructurada y acorde al problema planteado.
- La utilidad de utilizar y adaptar las metodologías ágiles a las necesidades del equipo.



Para concluir al equipo de TechResQ en el transcurso de desarrollo de este proyecto informático le ha ayudado a reconocer como abordar distintas problemáticas relacionadas al desarrollo de soluciones involucradas en patrocinadores, como también comprender los distintos pasos necesarios para desarrollar un proyecto de software, reforzando los aprendizajes obtenidos en el transcurso de nuestra carrera universitaria, como también integrando conocimientos relacionados al mundo laboral, concretando en una solución autentica, en un MVP funcional, capaz de facilitar problemas verdaderos y existentes en nuestra sociedad.

11 Anexos

- <https://asana.com/es/resources/scrum-master>
- <https://asana.com/es/resources/product-owner>
- <https://asana.com/es/resources/what-is-kanban>
- <https://donetonic.com/es/que-son-los-sprints-en-scrum>
- <https://kanbantool.com/es/guia-kanban/elementos-de-kanban>
- <https://www.pmi.org/disciplined-agile/agile/theagilemanifesto>
- <https://sentry.io/blog/valores-principios-agile-manifiesto-agil>
- <https://www.esic.edu/rethink/tecnologia/manifiesto-agile-valores-12-principios-c>
- <https://www.atlassian.com/es/agile/scrum>
- <https://agilemanifesto.org/iso/es/principles.html>
- <https://lucid.co/es/blog/diferentes-tipos-de-reuniones-de-scrum>
- <https://kelea.es/cual-es-la-diferencia-entre-los-equipos-autoorganizados-y-los-equipos-autogestionados>
- <https://www.linkedin.com/pulse/importance-collaboration-agile-methodologies-pgx2f>
- <https://devrev.ai/blog/agile-sprint-collaboration>
- <https://plaky.com/learn/project-management/scope-creep-in-project-management/>