



Tecnológico de Monterrey

TC3004B.102 Planeación de sistemas de software

Administración del proyecto

Cristóbal Alberto Escamilla Sada	A00827074
Fernando Doddoli Lankenau	A00827038
Aldo Beráin Cárdenas	A00827874
Juan Pablo Yáñez González	A00829598
Samuel Vieira Restrepo	A00828215

Profesora: Teresa de Jesús Lucio Nieto

18 de Marzo del 2022

Semestre Febrero-Junio 2022

Tec de Monterrey, Campus Monterrey

Tabla de Contenidos

Alcance del proyecto	2
Entregables del proyecto	2
Alcance del producto	2
Requerimientos de software	2
Work Breakdown Structure (WBS) del producto	3
Matriz RACI	4
Cronograma de actividades	5
Organización en Azure DevOps	5
Personas involucradas	8
Personas impactadas por el proyecto	8
Usuario final	8
Diagrama de Stakeholders	9
Comunicación	10
Gestión de riesgos	13
Plan de calidad	15
Métricas de calidad	15
Control de calidad	16
Plan de mejoras	16
Tabla de estimaciones	17

Alcance del proyecto

Este proyecto se realizará durante un periodo de quince semanas (14 de febrero - 17 de junio) y el trabajo será distribuido entre un equipo conformado por cinco personas. Las primeras cinco semanas de trabajo (14 de febrero - 18 de marzo) consisten en la planeación detallada del proyecto, incluyendo planes de trabajo, diagramas de administración mediante *Azure DevOps*, asignación de roles, documentación de requerimientos, documentación de planes de pruebas y un prototipo del sistema desarrollado con alta fidelidad. Las últimas diez semanas de trabajo (28 de marzo - 17 de junio) consistirán en una etapa de desarrollo de software. Durante este periodo se llevarán a cabo la programación del sistema, la fase de pruebas y el despliegue final. Se espera que el equipo de desarrollo trabaje a tiempo completo para producir una entrega de calidad.

Entregables del proyecto

A través de este proyecto, la empresa *Inflection Point Systems* recibirá una serie de entregables presentados a continuación:

- Documentación de requerimientos de software
- Prototipo completo del software en alta fidelidad
- Documento de resultados de pruebas
- Código fuente del sistema implementado
- Manual de instalación del software
- Manual de uso del software

Alcance del producto

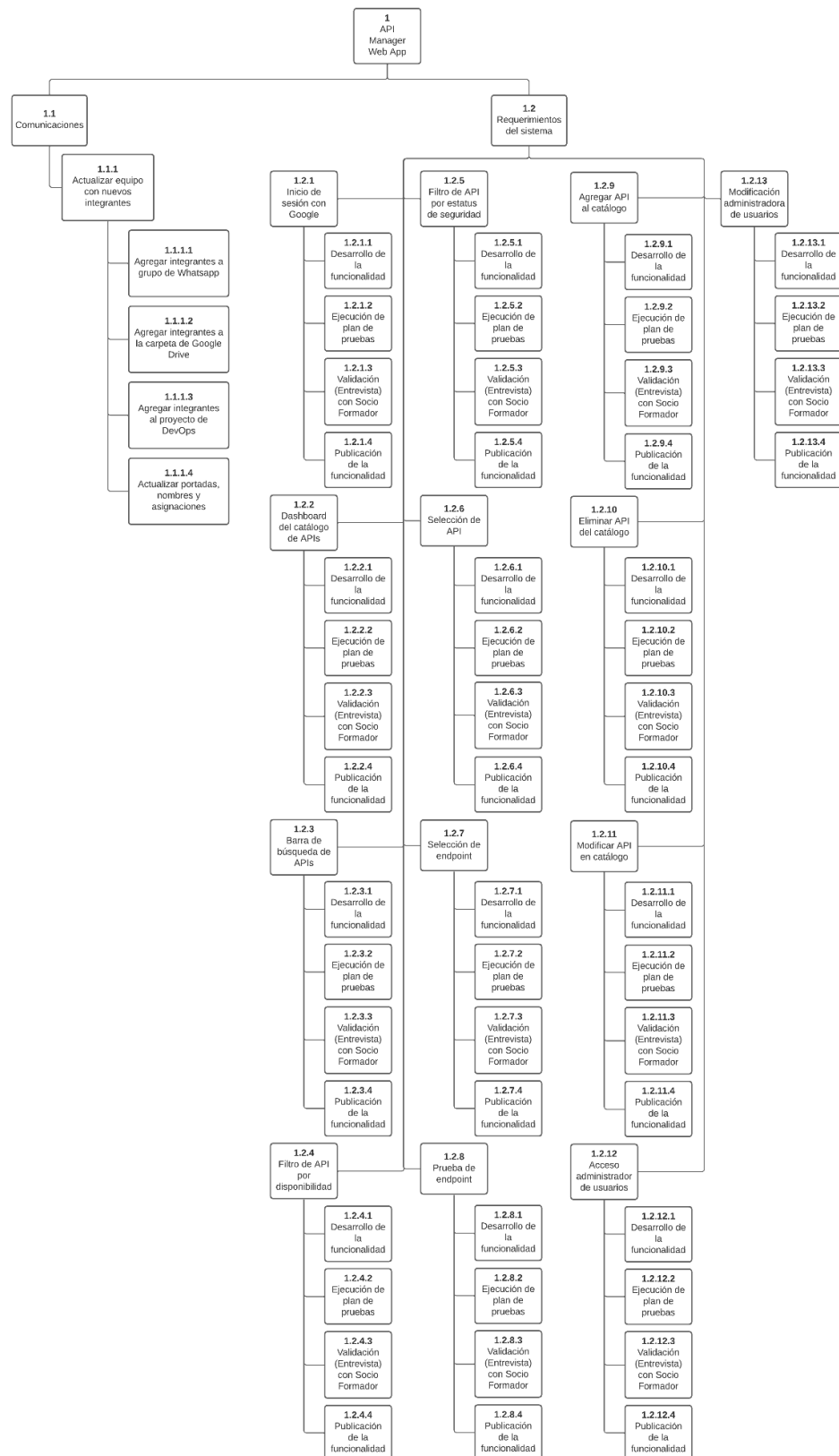
A través de una aplicación web, se implementará un catálogo de APIs para el uso de los empleados desarrolladores y administradores de *Inflection Point Systems*. Este contará con la lista de APIs disponibles junto con su información de uso. Dicha información incluye: URL de invocación, endpoints, descripción, parámetros de entrada y formato de respuesta. Es importante mencionar que estos APIs publicados podrán ser probados por los usuarios desarrolladores y administradores ingresando los parámetros requeridos del respectivo endpoint a probar. El usuario desarrollador también deberá poder ver el status del API con un sistema de semáforos cual represente dicho estatus con un respectivo color. El usuario administrador tendrá el privilegio de agregar, modificar y eliminar APIs del catálogo.

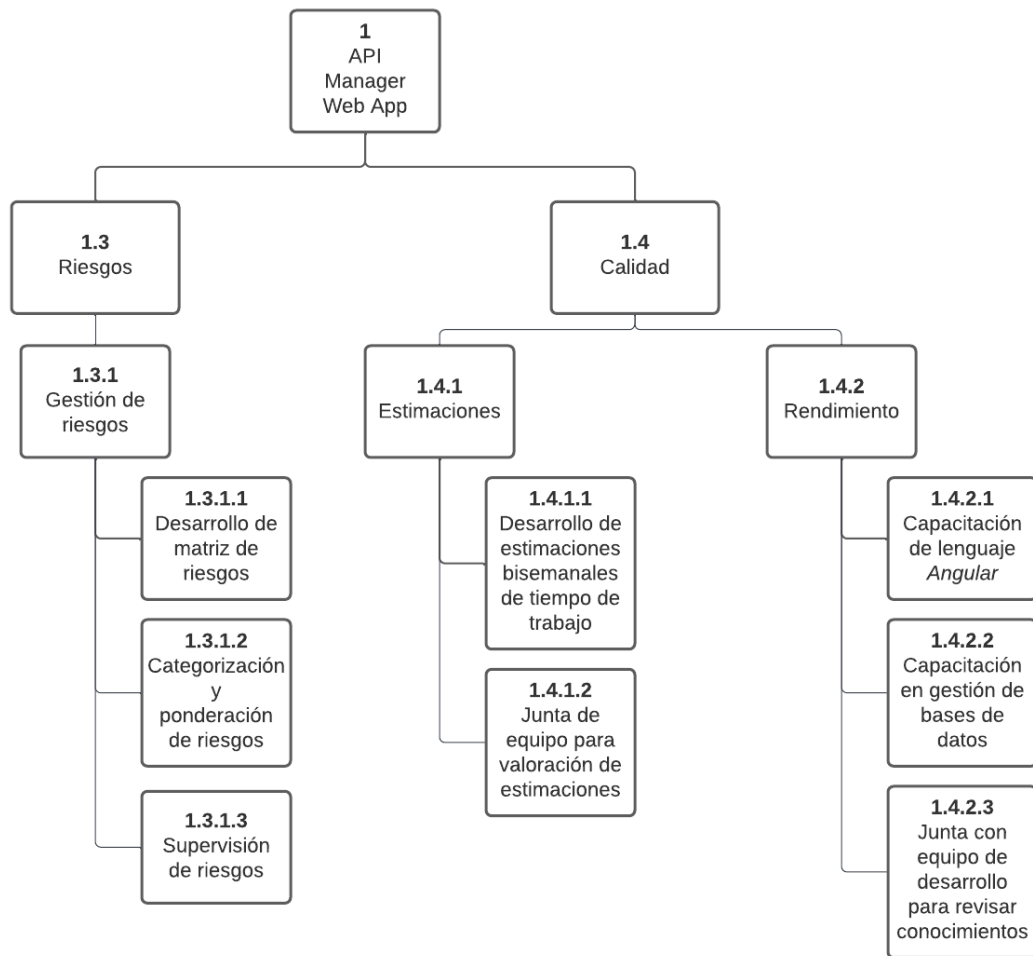
Requerimientos de software

Se realizó una serie de entrevistas con el socio formador para la elicitación de requerimientos. Posteriormente, se realizó un documento incluyendo trece requerimientos que se desarrollarán durante las próximas diez semanas.

Work Breakdown Structure (WBS) del producto

Este diagrama de WBS contiene los trabajos a realizar durante las últimas diez semanas del proyecto. El diagrama está dividido en las siguientes dos imágenes:



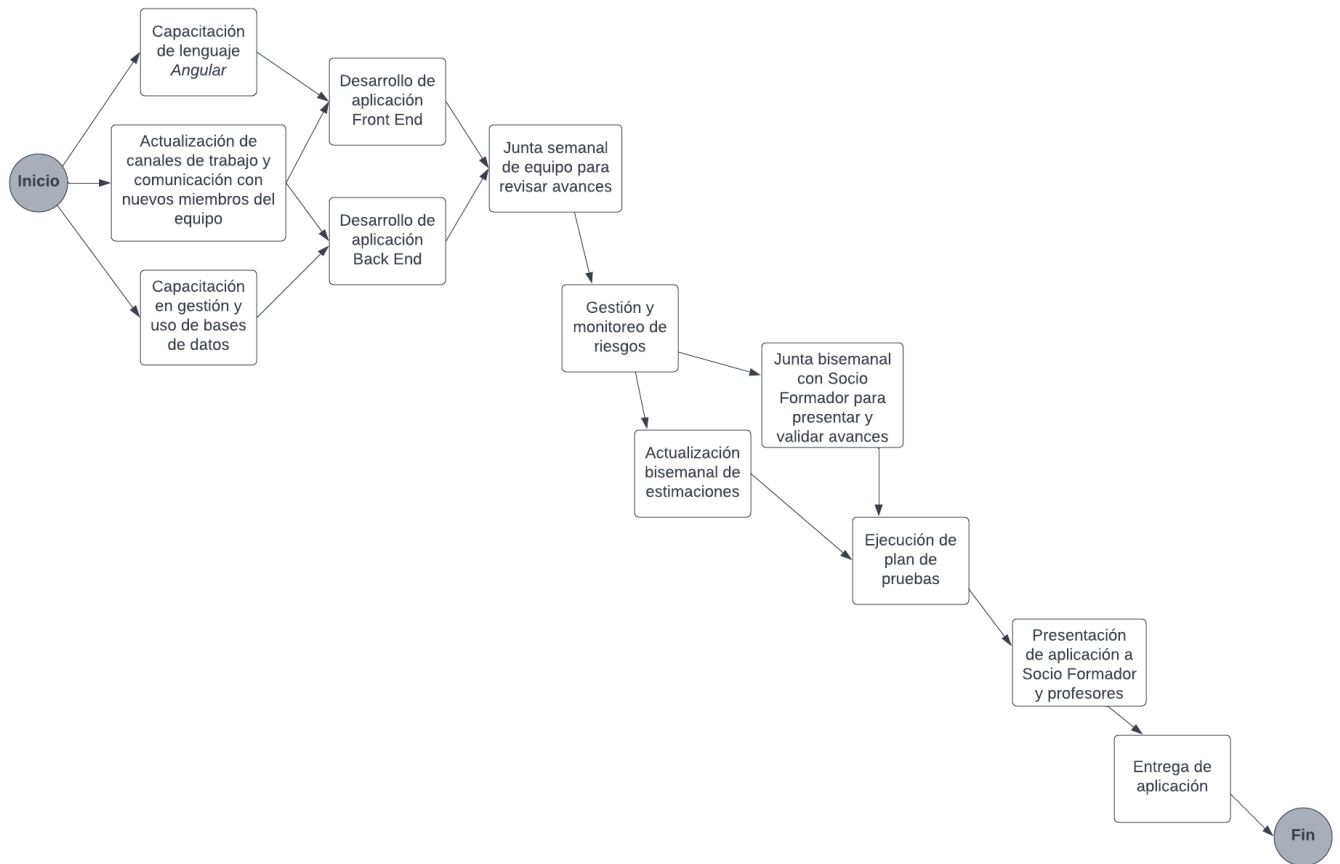


Matriz RACI

Matriz RACI					
Tareas del proyecto	Cristóbal Escamilla	Fernando Doddoli	Samuel Vieira	Aldo Beráin	Juan pablo Yáñez
Tareas del proyecto	Lider de Proyecto	Desarrollador de UI	Desarrollador Back End	Gestor de riesgos	Control Calidad y Beta tester
Fase inicio					
Creación de proyecto en Azure DevOps	A	C	R	I	I
Canales de comunicación	R	A	A	A	A
Documentación inicial	R	A	A	A	A
Planeación					
Documentación de roles	R	C	C	C	C
Planeación de entrevistas con Socio	R	C	C	C	C
Elicitación de requerimientos	A	R	R	R	R
Documentación de requerimientos	A	I	I	C	R
Plan de pruebas	A	I	R	C	I
Documentación de riesgos	A	C	C	R	C
Elaboración de prototipo	A	R	C	I	I
Entrega de planeación	R	A	A	A	A
Desarrollo					
Desarrollo de UI	A	R	C	I	C
Desarrollo de BackEnd	A	C	R	I	C
Control de versiones	A	C	C	I	R
Fase de Control					
Ejecución de pruebas	A	R	R	C	C
Fase de Cierre					
Manual de uso	R	C	C	I	A
Manual de instalación	A	C	R	A	I
Implantación del software	A	R	C	I	A
Entrega final	R	A	A	A	A

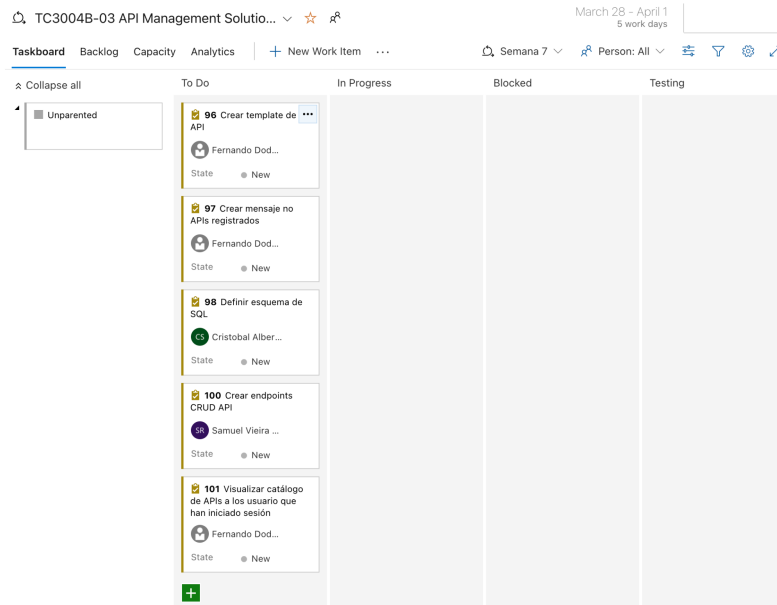
Legenda
Responsable
Accountable
Consulted
Informed

Cronograma de actividades

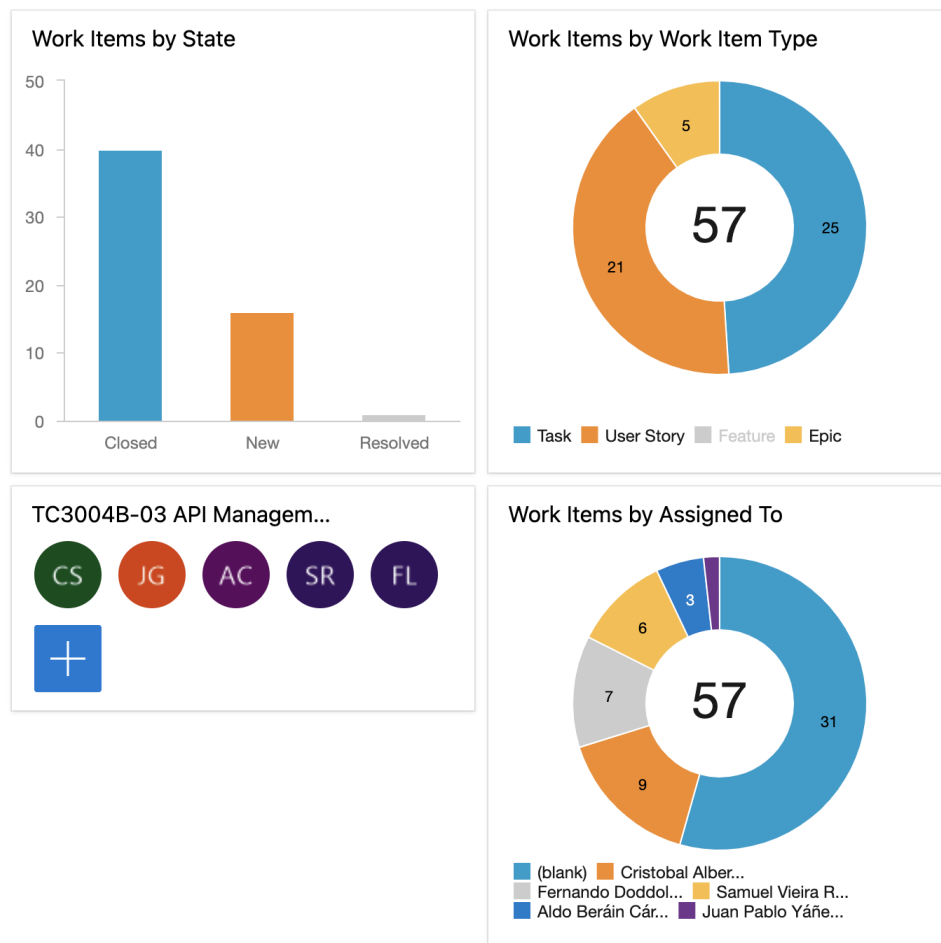


Organización en Azure DevOps

Durante las primeras cinco semanas nos dedicamos a la planificación de nuestro plan de desarrollo para las próximas diez semanas. En este bloque seguimos una metodología de desarrollo ágil, donde nos estaremos enfocando en completar pequeñas partes de nuestro proyecto a la vez. Dada la naturaleza de esta metodología de desarrollo, hemos planeado el primer sprint de nuestro proyecto, donde estaremos desarrollando la interfaz básica de la aplicación.

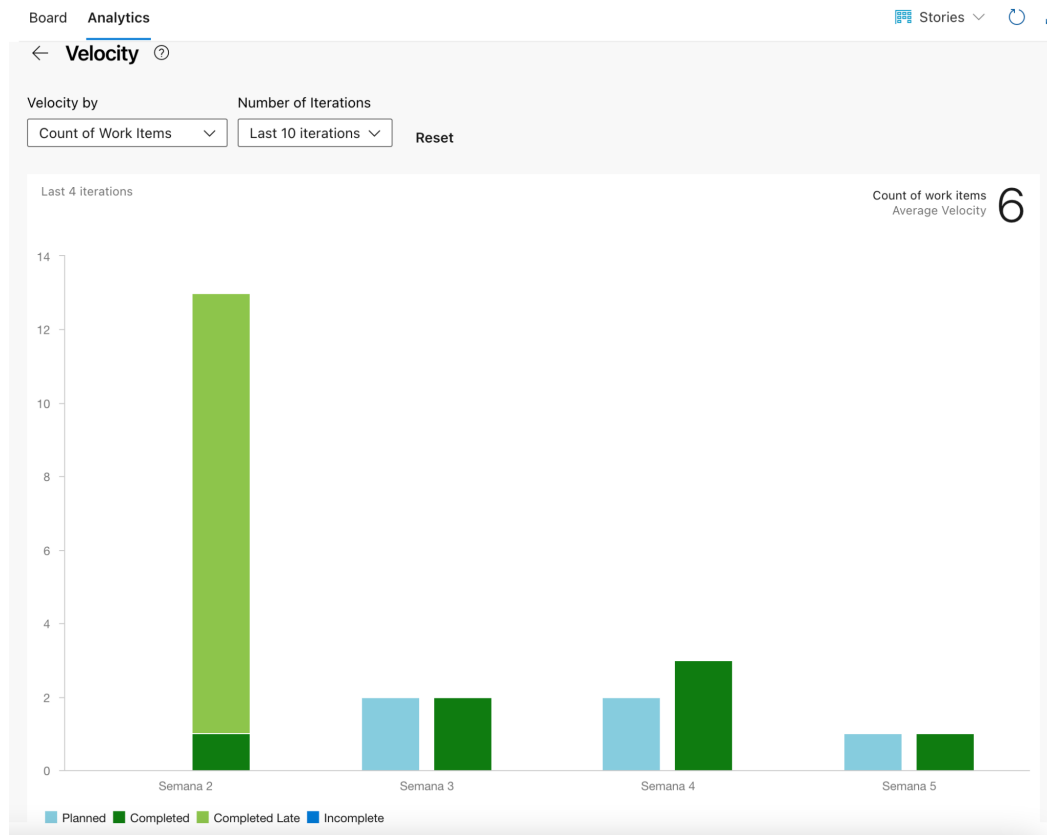


Antes de comenzar este sprint, nuestro backlog consiste de cinco tasks, las cuales hacen parte de la feature *Presentación en renglones*, y *Bases de datos* respectivamente.

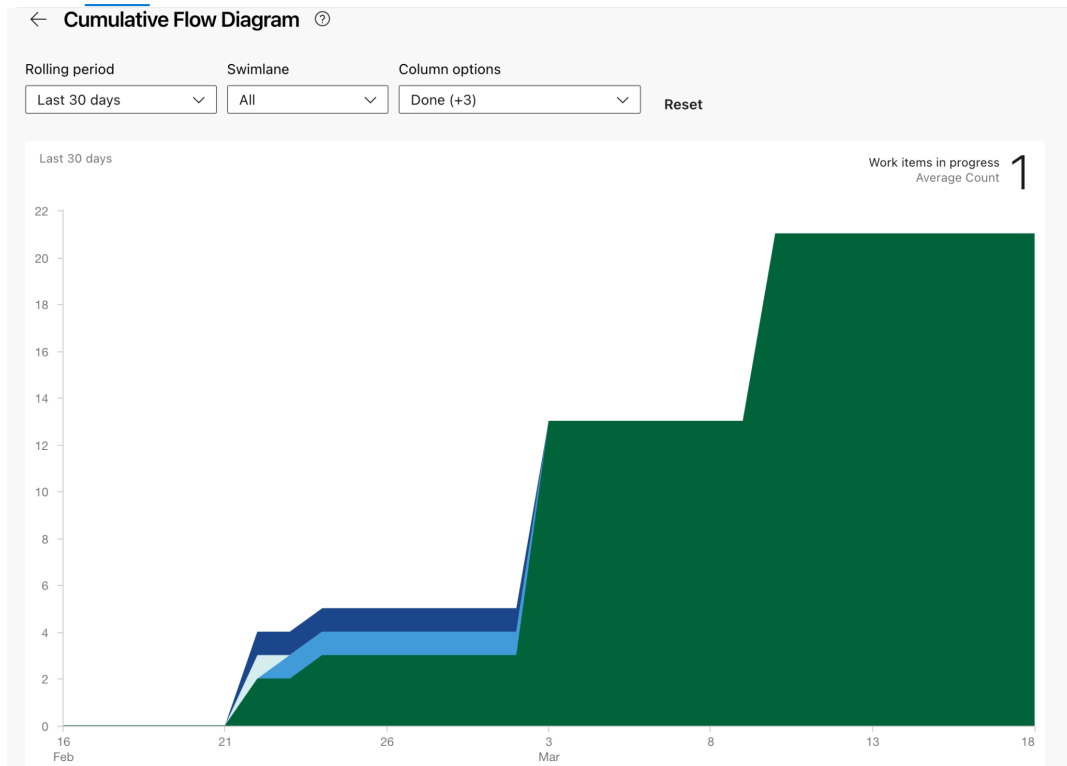


A lo largo de estas primeras cinco semanas el tipo de planeación ha cambiado. En el comienzo estuvimos planeando temas fundamentales del proyecto que estábamos planeando. Mientras este ha ido tomando forma poco a poco, nos hemos comenzado a enfocar en temas

más puntuales de desarrollo. Este cambio en enfoque ha generado una transición de historias de usuario y Epics a Features y tasks directas. Esto significa que nos estamos acercando al desarrollo técnico de la plataforma.



En el diagrama de flujo acumulativo podemos ver que la cantidad de tareas que están siendo completadas ha incrementado en comparación con las semanas anteriores. Sin embargo, este también nos muestra que los elementos están siendo transferidos directamente de *pendiente* a *terminado*. Esto nos indica que hay que generar capacitación dentro del equipo en cuanto al uso de los boards de Azure.



Personas involucradas

Actualmente contamos con tres involucrados importantes, los cuales son nuestro socio formador Inflexion point, los profesores y finalmente los desarrolladores.

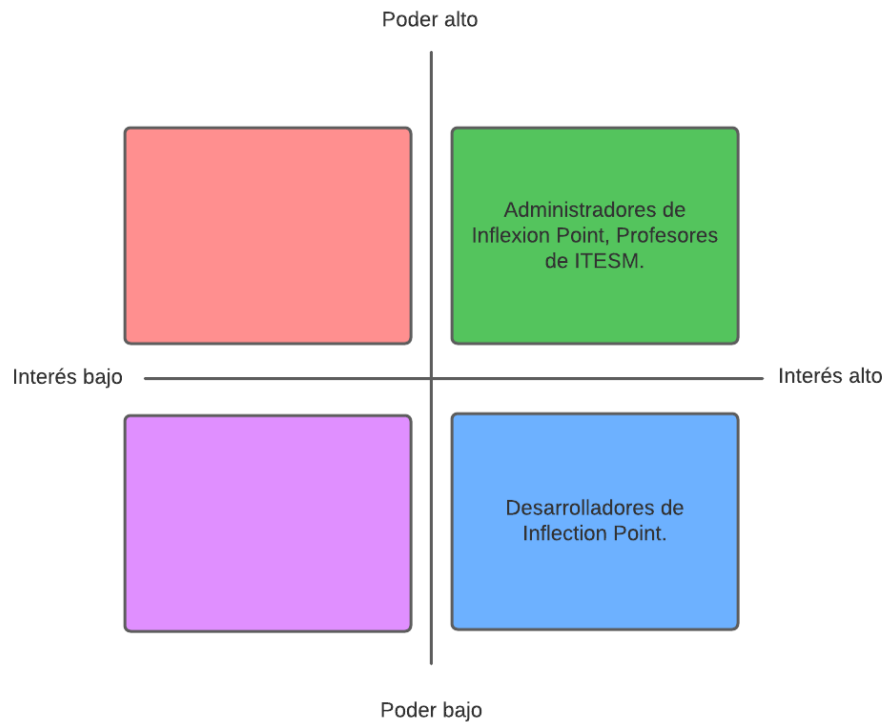
Personas impactadas por el proyecto

Las personas que sufrirán algún impacto por el proyecto serían todos los interesados, a los maestros les afectaría en su relación con la empresa, a nosotros en nuestra calificación y a inflection point porque no saldría su sistema.

Usuario final

El usuario final del proyecto serán los empleados desarrolladores y administradores de Inflexion Point, los cuales consultarán la aplicación para obtener información sobre alguna API.

Diagrama de *Stakeholders*



En nuestro análisis de interesados podemos ver que solo contamos con personas a favor de este proyecto en este momento, lo cual es una ventaja ya que no requerimos disminuir el impacto de las personas en contra, además de esto todas las personas que apoyan el proyecto tienen un gran interés.

Comunicación

Tabla de contactos		
Integrante del equipo	Correo Electrónico	Teléfono Celular
Aldo Beráin Cárdenas	A00827874@tec.mx	+52 87 7788 1316
Fernando Doddoli Lankenau	A00827038@tec.mx	+52 81 1544 3495
Cristóbal Alberto Escamilla Sada	A00827074@tec.mx	+52 81 1990 2303
Samuel Vieira Restrepo	A00828215@tec.mx	+52 81 3590 9219
Juan Pablo Yáñez González	A00829598@tec.mx	+52 77 7565 3681

Tabla de juntas					
Aspecto a comunicar	Cuándo	Destinatario	Responsables	Canales	Tecnologías
Avances del proyecto	Dos veces por semana, durante 10 semanas.	Administradores de Inflexion Point y profesores ITESM.	Aldo Beráin Cárdenas Fernando Doddoli Lankenau Cristóbal Alberto Escamilla Sada Samuel Vieira Restrepo Juan Pablo Yáñez González (Equipo de desarrollo)	Presencial de ser posible. Medios digitales.	Zoom
Dudas sobre el proyecto.	Cada que se presente alguna.	Profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom

Logros del equipo.	Dos veces por semana, durante 10 semanas.	Administradores Inflexion Point y profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom
Problemas pendientes y resueltos.	Dos veces por semana, durante 10 semanas.	Administradores Inflexion Point y profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom
Estado de riesgos y su plan de mitigación.	Una vez por semana, durante 10 semanas.	Administradores Inflexion Point y profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom
Solicitudes de apoyo.	Cuando se necesita alguna ayuda del socio formador para la continuación del proyecto.	Administradores Inflexion Point.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom
Milestones y actividades siguientes.	Dos veces por semana, durante 10 semanas.	Administradores Inflexion Point y profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom
Costo del proyecto.	Dos veces por semana, durante 7 semanas.	Administradores Inflexion Point y profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom
Presentación MVP	Algún día de la cuarta semana.	Administradores Inflexion Point y profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom
Causas de la situación actual.	Dos veces por semana, durante 5 semanas.	Administradores Inflexion Point y profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom
Calidad actual del	Dos veces por semana,	Administradores Inflexion Point y	Equipo de desarrollo	Presencial de ser posible.	Zoom

proyecto y sus secciones de mejora.	durante 4 semanas.	profesores ITESM.		Medios digitales.	
Plan de mejoras de calidad.	Dos veces por semana, durante 3 semanas.	Administradores Inflexion Point y profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom
MBI 1.1	Algún día de la octava semana.	Administradores Inflexion Point y profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom
MBI 1.2	Algún día de la novena semana.	Administradores de Inflexion Point y profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom
Final del proyecto.	Algún día de la quinta semana.	Administradores Inflexion Point y profesores ITESM.	Equipo de desarrollo	Presencial de ser posible. Medios digitales.	Zoom

Tabla de responsabilidades comunicativas	
Responsabilidad	Responsable
Gestionar las comunicaciones ascendentes.	Aldo Beráin Cárdenas
Planificar la gestión de comunicaciones.	Fernando Doddoli Lankenau
Planear y organizar juntas de comunicación.	Cristóbal Alberto Escamilla Sada
Gestionar y transmitir las comunicaciones descendentes.	Samuel Vieira Restrepo
Gestionar y transmitir las comunicaciones horizontales.	Juan Pablo Yáñez González

Gestión de riesgos

ID	RIESGO	WBS	Contexto	Prob. Ocurre	Impacto	Factor	Nivel	Controles	Estatus
RS-01	Que no funcione la autenticación a través de Google	Inicio de sesión con Google	La única manera para poder ingresar a la plataforma es a través de Google	2	5	10	Importante	Tratar de implementar a futuro alguna otra manera de poder hacer login a la plataforma para no solo depender de una manera ya que al depender solo de 1, y esta llega a fallar la plataforma queda inoperable	Se identificó el riesgo: 15/03/2022
RS-02	Caída de Azure DevOps	Administración del proyecto	La planeación del desarrollo de software es llevada a cabo a través de Azure DevOps. Por lo tanto si este falla, los desarrolladores no podrán visualizar cosas importantes dentro del desarrollo del proyecto.	2	4	8	Apreciable	Prepararnos con una plan B para poder llevar a cabo el desarrollo del proyecto como canales de comunicación y estar al tanto de cómo está el proyecto para no solo depender de Azure	Se identificó el riesgo: 15/03/2022
RS-03	Falta de capacitación del equipo de desarrollo	Capacitación de lenguaje Angular	El sistema se desarrollará utilizando el lenguaje de programación <i>Angular</i> . Ninguno de los cinco miembros tiene experiencia utilizando dicho lenguaje.	2	5	10	Importante	Asistir a clases de capacitación en <i>Angular</i> e implementar un sistema de autoestudio en el equipo para de esta manera dominar el lenguaje	Se identificó el riesgo: 15/03/2022
RS-04	No poder visualizar los APIs registrados en el catálogo.	Dashboard del catálogo de APIs	La página estará conectada a una base de datos de donde consumirá toda la información a desplegar en el catálogo	3	5	15	Muy grave	Implementar algo que pueda mostrar la última versión de la base de datos para cuando esta no esté funcionando, mínimo se pueda visualizar las APIs que fueron dadas de alta	Se identificó el riesgo: 15/03/2022
RS-05	Pérdida de comunicación con el socio formador	Administración del proyecto	El socio formador es nuestro cliente, por lo que si se pierde comunicación constante con ellos puede que la aplicación no cumpla con sus necesidades.	2	5	10	Importante	Tener formas de contactarnos con el socio formador en caso de que se ocupe (correo electrónico, teléfono de Infection Point, ubicación de Infection Point)	Se identificó el riesgo: 15/03/2022
RS-06	Un integrante del equipo se enferma de covid.	Equipo de trabajo	El equipo está conformado por 5 integrantes, por lo que si una persona se enferma el tiempo de planeado de desarrollo puede ser severamente afectado.	2	3	6	Apreciable	Cuidarnos durante el transcurso del proyecto para evitar contagios innecesarios y estar preparados/capacitados para poder cubrir a alguien	Se identificó el riesgo: 15/03/2022
RS-07	No tener el proyecto terminado antes de la fecha de entrega acordada con el Socio Formador	Desarrollo del proyecto	Tenemos un tiempo establecido para desarrollar el software. Si no lo tenemos listo antes de la fecha acordada, no cumpliremos con la expectativa del cliente.	4	5	20	Muy grave	No subestimar las entregas y gestionar de una mejor manera el proyecto a través de Azure DevOps	Se identificó el riesgo: 15/03/2022
RS-08	Que por algún motivo, algún integrante del equipo deje de ser parte del equipo	Equipo de trabajo	El equipo está conformado por 5 integrantes, por lo que si una persona deja de ser parte del equipo, ahora los nuevos integrantes no conocerán al 100% el proyecto y forma de trabajar lo cual será un reto	5	5	25	Muy grave	Ir preparando medidas para poder hacer que el golpe de que se vayan integrantes y lleguen nuevos, sea lo más suave posible	Se identificó el riesgo: 15/03/2022
RS-09	El socio formador nos cambia los requerimientos a última hora	Comunicación con los stakeholders	El proyecto es creado en base a los requerimientos asignados por el socio formador, por lo cual si el socio formador decide cambiar los requerimientos a última hora, el proyecto estaría incompleto	2	5	10	Importante	Mantenernos comunicados con el socio formador lo más posible para que siempre estén al tanto con nuestros avances para asegurar que sea lo que quieren.	Se identificó el riesgo: 15/03/2022
RS-10	Las respuestas del socio formador son incongruentes con la funcionalidad de la aplicación	Comunicación con los stakeholders	El socio formador es nuestro cliente, por lo que si no hay buena comunicación puede que la aplicación no cumpla con sus necesidades.	2	4	8	Apreciable	Mantenernos comunicados con el socio formador lo más posible para que siempre estén al tanto con nuestros avances para asegurar que sea lo que quieren.	Se identificó el riesgo: 15/03/2022
RS-11	Que el producto final sea diferente a los prototipos de alta fidelidad	Desarrollo del proyecto	Como desarrolladores presentamos prototipos de alta fidelidad a los socioformadores los cuales fueron aprobados y esperan que el producto final sea así.	3	4	12	Importante	Ir estudiando de fondo al framework a utilizar para poder hacer que el proyecto sea lo más acertado posible a los prototipos de alta fidelidad	Se identificó el riesgo: 15/03/2022
RS-12	Baja calidad de código	Calidad de Software	Como miembros del equipo seremos responsables de codificar front end y back end del proyecto en Angular y C# respectivamente	3	4	12	Importante	Empezar a ver tutoriales e investigar y documentarnos acerca de las buenas prácticas para desarrollar con software de calidad con código de calidad	Se identificó el riesgo: 15/03/2022
RS-13	Cantidad baja de pruebas de software	Calidad de Software	Una de las prácticas importantes y que muchos desarrolladores no lo llevan a cabo de manera correcta son las pruebas de software. Tenemos que realizar las pruebas suficientes para probar que todo esté bien	2	5	10	Importante	Investigar y establecer qué tipo de pruebas vamos a realizar para tratar de cubrir y realizar pruebas a todo el software para corroborar que no haya ninguna falla	Se identificó el riesgo: 15/03/2022
RS-14	El proyecto no cubre las necesidades del socio formador	Comunicación con los stakeholders	El proyecto en el que estamos trabajando es para cubrir la necesidad de nuestro cliente, no lo que nosotros creamos que es la necesidad del cliente.	4	5	20	Muy grave	Es importante siempre mantener una buena comunicación con el socio formador para no perder el scope del proyecto y que sepan bien los avances que llevamos y cambios que hacemos para ver si seguimos todos en una misma página	Se identificó el riesgo: 15/03/2022
RS-15	Poca escalabilidad y mantenibilidad del proyecto	Desarrollo del proyecto	Pudiera ser que el socio formador llegara a querer agregar más funcionalidades al sistema y de no hacer el proyecto de manera correcta esto tal vez no sería posible y si sí, sería muy costoso	3	3	9	Importante	Llevar a cabo buenas prácticas a la hora de programar el sistema para que no sea difícil agregar funcionalidades y para que futuros miembros del equipo no se les dificulte entender el código	Se identificó el riesgo: 15/03/2022

LEYENDA							
			GRAVEDAD (IMPACTO)				
			MUY BAJO 1	BAJO 2	MEDIO 3	ALTO 4	MUY ALTO 5
PROBABILIDAD	MUY ALTA	5	5	10	15	20	25
	ALTA	4	4	8	12	16	20
	MEDIA	3	3	6	9	12	15
	BAJA	2	2	4	6	8	12
	MUY BAJA	1	1	2	3	4	5

	Riesgo muy grave. Requiere medidas preventivas urgentes. No se debe iniciar el proyecto sin la aplicación de medidas preventivas urgentes y in acotar sólidamente el riesgo.
	Riesgo importante. Medidas preventivas obligatorias. Se deben controlar fuertemente las variables de riesgo durante el proyecto.
	Riesgo apreciable. Estudiar económicamente si es posible introducir medidas preventivas para reducir el nivel de riesgo. Si no fuera posible, mantener las variables controladas.
	Riesgo marginal. Se vigilará aunque no requiere medidas preventivas de partida.

Código	Riesgo	Descripción
RS-01	Que no funcione la autenticación a través de Google	La única manera para poder ingresar a la plataforma es a través de Google
RS-02	Caída de Azure DevOps	La planeación del desarrollo de software es llevada a cabo a través de Azure DevOps. Por lo tanto si este falla, los desarrolladores no podrán visualizar cosas importantes dentro del desarrollo del proyecto.
RS-03	Falta de capacitación del equipo de desarrollo	El sistema se desarrollará utilizando el lenguaje de programación Angular. Ninguno de los cinco miembros tiene experiencia utilizando dicho lenguaje.
RS-04	No poder visualizar los APIs registrados en el catálogo.	La página estará conectada a una base de datos de donde consumirá toda la información a desplegar en el catálogo
RS-05	Pérdida de comunicación con el socio formador	El socio formador es nuestro cliente, por lo que si se pierde comunicación constante con ellos puede que la aplicación no cumpla con sus necesidades.
RS-06	Un integrante del equipo se enferma de covid.	El equipo está conformado por 5 integrantes, por lo que si una persona se enferma el tiempo de planeado de desarrollo puede ser severamente afectado.
RS-07	No tener el proyecto terminado antes de la fecha de entrega acordada con el Socio Formador	Tenemos un tiempo establecido para desarrollar el software. Si no lo tenemos listo antes de la fecha acordada, no cumpliremos con la expectativa del cliente.
RS-08	Que por algún motivo, algún integrante del equipo deje de ser parte del equipo	El equipo está conformado por 5 integrantes, por lo que si una persona deja de ser parte del equipo, ahora los nuevos integrantes no conocerán al 100% el proyecto y forma de trabajar lo cual será un reto
RS-09	El socio formador nos cambia los requerimientos a última hora	El proyecto es creado en base a los requerimientos asignados por el socio formador, por lo cual si el socio formador decide cambiar los requerimientos a última hora, el proyecto estaría incompleto
RS-10	Las respuestas del socio formador son incongruentes con la funcionalidad de la aplicación	El socio formador es nuestro cliente, por lo que si no hay buena comunicación puede que la aplicación no cumpla con sus necesidades.
RS-11	Que el producto final sea diferente a los prototipos de alta fidelidad	Como desarrolladores presentamos prototipos de alta fidelidad a los socioformadores los cuales fueron aprobados y esperan que el producto final sea así.

RS-12	Baja calidad de código	Como miembros del equipo seremos responsables de codificar front end y back end del proyecto en Angular y C# respectivamente
RS-13	Cantidad baja de pruebas de software	Una de las prácticas importantes y que muchos desarrolladores no lo llevan a cabo de manera correcta son las pruebas de software. Tenemos que realizar las pruebas suficientes para probar que todo esté bien
RS-14	El proyecto no cubre las necesidades del socio formador	El proyecto en el que estamos trabajando es para cubrir la necesidad de nuestro cliente, no lo que nosotros creamos que es la necesidad del cliente.
RS-15	Poca escalabilidad y mantenibilidad del proyecto	Pudiera ser que el socio formador llegara a querer agregar más funcionalidades al sistema y de no hacer el proyecto de manera correcta esto tal vez no sería posible y si sí, sería muy costoso

Plan de calidad

La calidad de software es sinónimo de eficiencia, flexibilidad, confiabilidad, mantenibilidad, portabilidad, entre otras características importantes que los softwares deben contar ya que todas estas cualidades determinan su utilidad y existencia. Para que un producto de software pueda ser considerado exitoso, el producto debe cumplir con estándares de calidad relacionados a la funcionalidad y rendimiento ajustado a las necesidades y exigencias del cliente. Debido a lo anteriormente mencionado, hemos establecido 2 medidas para asegurarnos que el software tenga la mejor calidad posible, que cumpla y de ser posible, que supere las expectativas de nuestros clientes. Las medidas definidas son:

- Establecer métricas para poder medir objetivamente la calidad del software.
- Evaluar el impacto en la calidad cada vez que cambia el alcance, tiempo, costo, recursos y riesgos.

Métricas de calidad

- Cumplimiento de historias de usuario definidas.
- Número de errores por cada historia de usuario.
- Cantidad de horas definidas y cantidad de horas trabajadas.
- Sesiones completas con el Socio Formador.
- Cumplimiento con estimaciones de tiempo

Control de calidad

Tabla semanal de calidad			
Item	Métrica	Resultado	
#		Sí	No
1	¿Se cumplieron los requerimientos planeados en la semana?	X	
2	¿Se cumplió la estimación de tiempos por tarea?	X	
3	¿Hubo junta con el socio formador?	X	
4	¿Se documentaron los cambios en el software?	X	
5	¿Se capacitó correctamente al equipo?		X
6	¿Los cinco integrantes del equipo asistieron a las sesiones de trabajo?	X	
7	¿Se gestionaron los riesgos del proyecto?	X	
8	¿Hubo comunicación con los <i>Stakeholders</i> ?	X	
9	¿Hubo cambios en el itinerario del equipo?		X
10	¿El proyecto va a tiempo?	X	

Plan de mejoras

Para poder desarrollar un producto de software de calidad, es necesario contar con un plan de mejoras para que durante el desarrollo del proyecto, podamos actuar ante actividades que no agreguen valor a nuestro proyecto con anticipación.

Acciones para evitar riesgos	Riesgos posibles evitados	Descripción de la acción
Capacitar al equipo para utilizar herramienta de control de versiones	- Que ocurran problemas a la hora del desarrollo del código por falta de comunicación	Al momento de tener un nuevo equipo, es importante capacitar sobre el uso de herramientas de control de versiones como GitHub ya que no se sabe la manera en la que han trabajado anteriormente y muchos no saben de grandes herramientas que existen que pueden llegar a facilitar el proyecto de manera importante
Fomentar el uso de Azure DevOps	- Que ocurran problemas a la hora de la gestión del proyecto por falta de comunicación - No cumplir con el fecha estimada de entrega	Existen muchas herramientas para desarrollar un software sin embargo es importante fomentar el uso de Azure DevOps para que así todo el equipo esté en la misma página y al tanto de los cambios que se han realizado
Capacitar al equipo de desarrolladores para que hagan pruebas de software	- Encontrarse con bugs a futuro - Una complejidad alta del código	Una de las problemáticas en desarrollo de software es que el proyecto realizado no cumple con las expectativas del cliente y gran parte es porque no se realizan pruebas de software y al realizar pruebas podemos evitarnos de no cumplir con las expectativas del cliente en muchos ámbitos
Capacitar al equipo de desarrolladores para que sigan un estándar de código	- Que futuros integrantes del equipo no se les dificulte entender el código	En el desarrollo de software es muy común que haya nuevos integrantes durante el proyecto y es por eso que seguir un estándar es importante para que puedan trabajar en el código sin gastar mucho tiempo en entenderlo
Una junta semanal	- Que algún miembro del equipo pierda avances del proyecto - Que algún miembro del equipo malentienda algún el propósito de un avance por parte de otro miembro	A lo largo del proyecto, debido a que los integrantes tienen diferentes roles, es muy probable que alguno de los integrantes pierda el enfoque en el proyecto. Por eso, en una junta semanal se pueden abordar muchos temas para llevar a cabo una mejor gestión del proyecto
Incluir gamificación en el proyecto	- Que algún miembro del equipo pierda el interés del proyecto - Que algún miembro del equipo no realice trabajo de calidad por falta de empeño	La gamificación es una de las técnicas más utilizadas los últimos años para que los empleados de una empresa, tengan una motivación extra para realizar su trabajo.

Tabla de estimaciones

*Tiempo medido en horas

Actividad	Cristóbal	Aldo	Samuel	Fernando	JP	Total Equipo
Semana 1						
Capacitación en lenguaje <i>Angular</i>	2	2	2	2	2	10
Capacitación en gestión y uso de bases de datos	2	2	2	2	2	10
Desarrollo Front End catálogo de APIs	0	0	2	4	4	10
Desarrollo Back end catálogo de APIs	4	4	2	0	0	10
Junta de validación con Socio Formador	1	1	1	1	1	5
Totales Semanales	9	9	9	9	9	45
Semana 2						
Capacitación en lenguaje <i>Angular</i>	2	2	2	2	2	10
Capacitación en gestión y uso de bases de datos	2	2	2	2	2	10
Desarrollo Front End login	0	0	1	2	2	5
Desarrollo Back End login	2	2	1	0	0	5
Desarrollo Front End filtros	0	0	2	4	4	10
Desarrollo Back end filtros	4	4	2	0	0	10
Junta de validación con Socio Formador	1	1	1	1	1	5
Totales Semanales	11	11	11	11	11	55
Semana 3						
Capacitación en lenguaje <i>Angular</i>	2	2	2	2	2	10
Capacitación en gestión y uso de bases de datos	2	2	2	2	2	10
Desarrollo Front End barra de búsqueda/nav	0	0	2	2	2	6
Desarrollo Back end barra de búsqueda/nav	4	4	2	0	0	10
Junta de validación con Socio Formador	1	1	1	1	1	5
Totales Semanales	9	9	9	7	7	41
Semana 4						
Capacitación en lenguaje <i>Angular</i>	2	2	2	2	2	10
Capacitación en gestión y uso de bases de datos	2	2	2	2	2	10
Desarrollo Front End API details	2	2	2	4	4	14
Desarrollo Back end API details	4	4	2	2	2	14
Junta de validación con Socio Formador	1	1	1	1	1	5
Totales Semanales	11	11	9	11	11	53
Semana 5						
Capacitación en lenguaje <i>Angular</i>	2	2	2	2	2	10
Capacitación en gestión y uso de bases de datos	2	2	2	2	2	10
Desarrollo Front End API details	2	2	2	4	4	14
Desarrollo Back end API details	4	4	2	2	2	14
Junta de validación con Socio Formador	1	1	1	1	1	5
Totales Semanales	11	11	9	11	11	53
Semana 6						
Desarrollo Front End prueba endpoints	0	0	3	6	6	15
Desarrollo Back end prueba endpoints	6	6	3	0	0	15
Desarrollo Front End agregar API	0	0	3	6	6	15
Desarrollo Front End agregar API	6	6	3	0	0	15
Junta de validación con Socio Formador	1	1	1	1	1	5
Totales Semanales	13	13	13	13	13	65
Semana 7						
Desarrollo Front End modificar API	0	0	2	4	4	10
Desarrollo Back End modificar API	4	4	2	0	0	10
Desarrollo Front End eliminar API	0	0	2	4	4	10
Desarrollo Back End eliminar API	4	4	2	0	0	10
Junta de validación con Socio Formador	1	1	1	1	1	5
Totales Semanales	9	9	9	9	9	45
Semana 8						
Desarrollo Front End administración de usuarios	0	0	3	6	6	15
Desarrollo Back end administración de usuarios	6	6	3	0	0	15
Retoques finales Front End	0	0	3	6	6	15
Retoques finales Back end	6	6	3	0	0	15
Junta de validación con Socio Formador	1	1	1	1	1	5
Totales Semanales	13	13	13	13	13	65
Semana 9						

Semana 10						
Documentación de proyecto	4	4	4	4	4	20
Implantación del software	10	10	10	10	10	50
Entrega final a Socio Formador	1	1	1	1	1	5
Totales Semanales	15	15	15	15	15	75
GRAN TOTAL	116 Horas	116 Horas	112 Horas	114 Horas	114 Horas	572 Horas