

1er Semestre 2022
viernes, 24 de junio de 2022



**Universidad
Nacional de
General
Sarmiento**

PROYECTO ÁGUILA

Docentes:

Juan Carlos Monteros

Francisco Simón Orozco de la Hoz

Leandro Dikenstein Hidalgo



GRUPO 3

Esteban Merlos

Ezequiel Ramallo

Felipe Corso

Matías Morales

Nicolás Díaz Velázquez

Vanesa Vera

– esteban-merlos33@outlook.com

– ezeramallo@gmail.com

– pipecorso79@gmail.com

– matias.agustin.morales@gmail.com

– nicolas2k199@gmail.com

– vanesa.vera1127@gmail.com

Índice

Introducción	3
Misión y Visión del Proyecto	3
Equipo y Roles	4
Plan de comunicaciones	5
Tecnologías utilizadas	5
Gestión del Proyecto	6
Metodología utilizada	6
Entregables	6
WBS (Work Breakdown Structure)	7
Diagramas	7
Diccionario WBS Final	8
Calendario	8
Métricas estimadas vs reales	9
Horas de Trabajo	9
Velocidad del Equipo	11
Riesgos	12
Manejo de bugs	13
Administración de cambios	13
Indicadores	14
Burndown Chart	14
Indicador de Evolución	15
Cobertura de la Prueba	15
Integraciones	16
Que brindamos	16
Planes de vuelos dependiendo el estado del mismo	16
Información específica de cada aeronave.	16
Historial de planes de vuelo.	17
Vuelos por tripulación.	17
Vuelos por modelo de avión.	17
Historial de Usuarios.	17
Carga destino de un vuelo	17
Que requerimos	17
Rutas:	17
Carga origen de un vuelo	17
Aeropuertos	18
Ruta Alternativa / Ruta Real	18
Información técnica de modelo de aeronave	18
Control de cabina	18
Emergencias en un vuelo	18
Mantenimiento de aeronaves	18

Insumos por vuelo	19
Integraciones con APIs externas	19
Open Meteo	19
Sistema de mapas leaflet	19
Retrospectiva	19
Inconvenientes surgidos y resoluciones	19
No respetamos al pie de la letra las ceremonias de Scrum.	19
No separar apropiadamente las historias de usuario de gran tamaño.	20
No tener al día los avances de las User stories.	20
Lecciones aprendidas	20
¿Qué volveríamos a hacer?	20
¿Qué NO volveríamos a hacer?	21
Conclusión	21
Conclusiones del producto	21
Conclusiones del proyecto	21
Repositorios	22

Introducción

Misión y Visión del Proyecto

La empresa **Special Flights S.A** solicitó un sistema de planificación, seguimiento, control, evaluación y mejora continua de operaciones de plan de vuelo eficientes de las rutas de acuerdo a los estándares de seguridad aeronáutica y las reglamentaciones existentes tanto internas en la compañía como las legales nacionales e internacionales. Para ello, fue necesario crear tres subsistemas: “Operaciones Terrestres, carga y despacho”, “**Planificación y Operaciones de Vuelo**” y “Operaciones en Cabina y Mantenimiento Aeronáutico”. Para nuestro equipo fue asignado el subsistema “Planificación y Operaciones de Vuelo”, el cual se encarga principalmente de la carga, almacenamiento, visualización y modificación de información sobre los planes y operaciones de vuelo.

A partir del escenario planteado, para este proyecto se solicitó además gestionar la creación de este sistema, para poder visualizar el nivel de calidad y organización del mismo a lo largo de cada *sprint*¹.

A fin de facilitar la lectura y comprensión del informe, al finalizar esta explicación se describe el equipo, la forma en que se realizaron las comunicaciones y las tecnologías utilizadas en el desarrollo del producto.

En la sección 2 se expone todo lo relacionado a la administración del proyecto, se describe la metodología seleccionada para el desarrollo del proyecto explicando cómo se adaptó a nuestro equipo de trabajo y se mencionan todos los sprints realizados. Además, con el fin de explicar y contrastar la planificación inicial con la final se profundiza sobre los sprints 1 y 5, donde se comparan las WBS², los calendarios, los tiempos estimados y lo efectivamente realizado, la gestión de riesgos, el manejo de bugs, la administración de cambios e indicadores.

Seguidamente, en la sección 3 se describen las integraciones con las que fue necesario trabajar, ya sea al realizarlas o utilizarlas.

Luego, en el apartado 4 se realiza una retrospectiva sobre todo la gestión del proyecto, haciendo hincapié en inconvenientes y lecciones aprendidas, incluyendo comentarios y conclusiones pertinentes a cada *sprint*.

Finalmente en la sección 5 se mencionan algunas conclusiones con respecto a la experiencia con la gestión de este proyecto y al producto final obtenido.

¹ Sprint es un ciclo de trabajo dentro de la metodología de trabajo Scrum, cuya duración estimada es de 2 a 4 semanas.

² Work Breakdown Structure(**WBS**) es la descomposición de un proyecto que está organizado en varios niveles.

Equipo y Roles

Al inicio del proyecto se asignaron roles principales y secundarios dentro del proyecto a cada miembro del equipo. Estas asignaciones se mantuvieron durante todo el proyecto.

Rol	Descripción	Responsable	
Product Owner	Encargado de la comunicación entre el cliente y el equipo Scrum.	Juan Carlos Monteros	
Scrum Master	Encargado de comunicar los requerimientos/cambios comunicados por el Product Owner o los distintos equipos ágiles en el proyecto. Además tiene la tarea de facilitar el trabajo de los miembros del equipo.	Nicolas Diaz Velazquez	
Desarrollador	Integrante del equipo de trabajo, se encarga de la implementación de las features.	Principales	Ezequiel Ramallo
			Felipe Corso
			Matias Morales
		Secundarios	Esteban Merlos
			Nicolas Diaz Velazquez
			Vanesa Vera
Tester	Integrante del equipo de trabajo, se encarga de la realización y ejecución de los test para las distintas funcionalidades a ser creadas.	Esteban Merlos Felipe Corso	
Lider Técnico	Se encarga de la correcta creación de los distintos documentos generados a lo largo del proyecto	Vanesa Vera	
Cliente	Persona o grupo que encargó el proyecto.	Special Flights S.A.	

Plan de comunicaciones

Además se eligieron tecnologías para la comunicación del equipo, se mantuvieron iguales que al inicio del proyecto.

Tipo de Comunicación	Partes	Canal de Comunicación	Canal Auxiliar
Entre Equipo de Trabajo	Todo el equipo	Telegram	Gmail/Email
		WhatsApp	
		Google Meet	Zoom
		Miro	
Entre Módulos	Scrum Master / Líder Técnico	Telegram	Gmail/Email
		Google Meet	Zoom
Con Product Owner	Scrum Master	Gmail / Email	Telegram

Tecnologías utilizadas

Con respecto a las tecnologías existieron algunos cambios, al inicio se escogió el motor de base de datos MySQL y la librería de Javascript JS Reports como herramienta de reportes, pero dado algunos inconvenientes que serán mencionados en el apartado 4, fueron reemplazados por el motor PostgreSQL y MetaBase, respectivamente.

Tipo	Tecnología	Descripción
Frontend	HTML	Lenguaje de etiquetado.
	CSS	Lenguaje de estilos.
	JS	Lenguaje de programación web.
Reportes	MetaBase	Creación de reportes
Backend	NODEJS	Entorno de backend.
	Express	Librería para la creación de la API REST
Servidor	Heroku	Deploy de servidor en la nube
Base de Datos	PostgresSQL	Motor de base de datos.
Testing	Postman	Software de testing para APIs
IDE	Visual Studio Code	Editor de código
Estimación	Planning Poker	Técnica de estimación

Documentación	Word/Google Docs	Creación de documentos
	Google Presentation	Creación de presentaciones
	Draw.io	Creación de diagramas
Calendarización	Proyect Libre	Creación de calendarios

Gestión del Proyecto

Metodología utilizada

La metodología principal utilizada por el equipo fue la metodología *Scrum*. Para este *framework incremental*, se hizo uso de listas de tareas pendientes denominadas *sprints*, que tuvieron una duración planificada y fija de 2 semanas. Al final de cada *sprint* se entregó un informe de avances y se realizó una retrospectiva entre los miembros del equipo para así mejorar los puntos para *sprints* posteriores. Además, el equipo realizó reuniones diarias denominadas *dailies* donde se realizó un seguimiento de los problemas ocurridos, las necesidades de cada integrante y las tareas que cada uno realizaría en el día. Por último, para aplicar esta metodología utilizamos la herramienta *Jira*, plataforma que ofrece múltiples herramientas que facilitan el trabajo de equipos ágiles para la gestión del proyecto.

Por otra parte, también se tomaron herramientas de la metodología *Scaled Agile Framework*. Esta metodología fue especialmente útil en la coordinación de equipos ágiles, ideal para la interacción entre los distintos equipos que trabajan en cada uno de los módulos del sistema. Las herramientas que se utilizaron fueron repositorios compartidos entre todos los equipos que permitieron tener un conocimiento del progreso actual de cada uno de los grupos.

Entregables

Los hitos, se mantuvieron desde el inicio del proyecto.

Tipo		Fecha
Presentación Inicial		22 de Abril
Reuniones Formales	Primera	26 de Abril
	Segunda	10 de Mayo
	Tercera	24 de Mayo

	Cuarta	7 de Junio
	Quinta	21 de Junio
Presentación Final		24 de Junio

WBS (Work Breakdown Structure)

Desde aquí se empiezan a observar los mayores cambios, al inicio del proyecto habíamos armado una WBS que contaba con 11 productos y 37 subproductos.

Para el final del proyecto, luego de todos los cambios, funcionalidades agregadas, fue necesario modificar la WBS que finalmente abarcó 22 productos y 81 subproductos, es decir, el doble de productos que al inicio. Estas modificaciones se relacionan con nuevos reportes o nuevas tablas que fueron necesarias agregar a lo largo del proyecto.

Diagramas

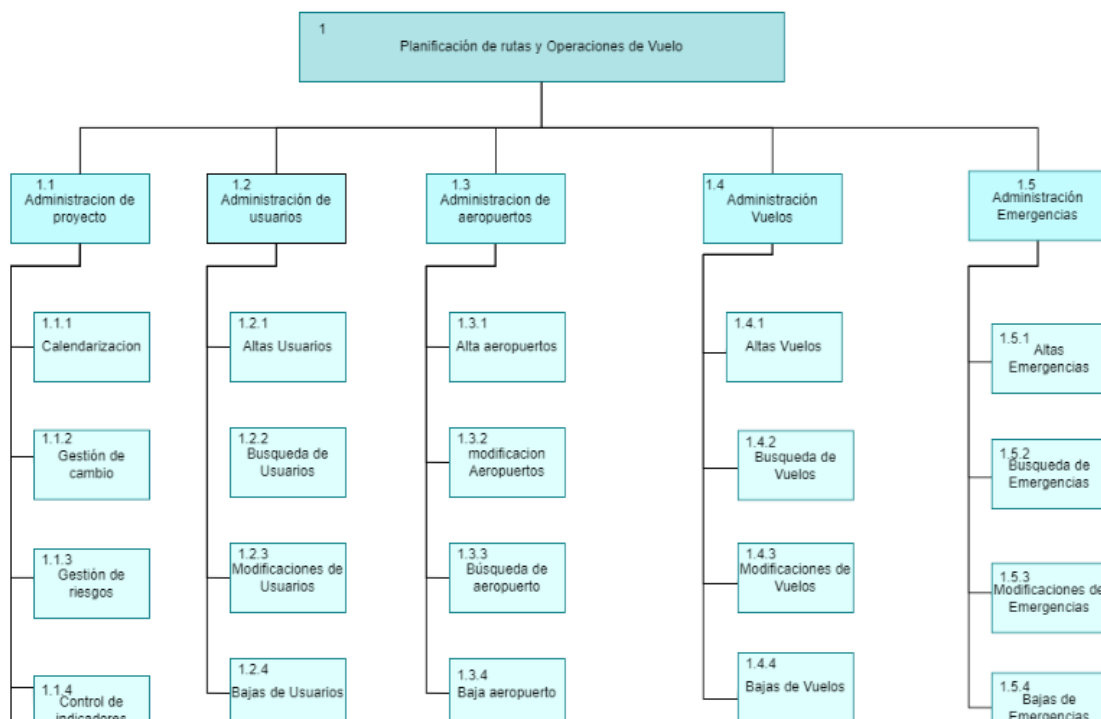


Figura 1. Captura de la WBS inicial, el diagrama completo se encuentra en el [anexo](#).

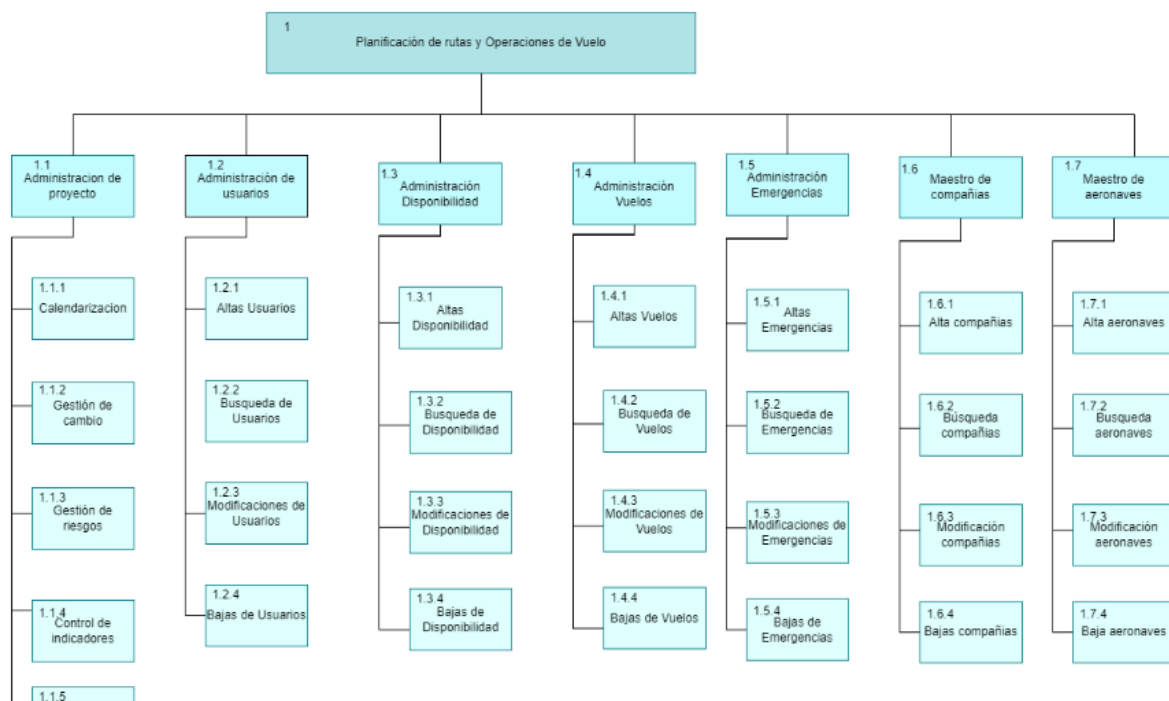


Figura 2. Captura de la WBS Final, el diagrama completo se encuentra en el [anexo](#).

Diccionario WBS Final

ID	Nombre	Descripción	Tarea	Esfuerzo	Responsable
1.1	Administración de proyecto	Planificación del proyecto	Contiene los siguientes subproductos: Calendarización, Gestión de cambio, Gestión de riesgos, Control de proyecto-indicadores, Gestión de las comunicaciones y Capacitación de nuevas tecnologías	320	Nicolás Díaz Velazquez,
1.1.1	Calendarización	Calendarización del proyecto. Marcar objetivos y fechas de inicio y fin	Secuenciar actividades, estimar recursos de actividad, estimar duración actividad y desarrollar el calendario.	40	Vanesa Vera

Figura 3. El diccionario completo se encuentra en el [anexo](#).

Calendario

Al inicio del proyecto se realizó un calendario basado en la WBS y en todas las funcionalidades que debían realizarse, el cual fue modificándose a lo largo del proyecto debido a cambios en las integraciones con otros módulos y nuevas funcionalidades.

A continuación se listan todas las funcionalidades se estimaron al inicio del proyecto para cada sprint y actividades extras que se realizaron dentro de este.

	Lo estimado	Lo agregado en lo real
Sprint 1	Plan de Proyecto Tentativo. Gestión Inicial. Capacitación tecnológica. Comunicación con módulos.	Re-cálculo de estimaciones.
Sprint 2	Capacitación tecnológica. Integración con módulos. Diseño de base de datos. Diseño de ABM de entidades identificadas.	Creación de los ABM de las entidades faltantes. Diseño del frontend.
Sprint 3	Modelado y creación de maestros Integración de servicios externos. Diseño e implementación de pantallas. Integración de backend - frontend. Creación de Reportes.	Diseño de reportes.
Sprint 4	Sincronización de base de datos. Integración Reportes	Implementación e integración con los servicios de los módulos del sistema. Capacitación nueva herramienta de reportes.
Sprint 5	Plan de Proyecto Final. Reportes Capacitación de usuario.	Refinamiento de funcionalidades. Detalles del frontend. Corrección de bugs y deuda técnica. Documentación, presentación, video de la presentación frente jurados.

Métricas estimadas vs reales

Horas de Trabajo

Como grupo se realizaron numerosas estimaciones en base a las horas de trabajo que tomaría cada sprint, para lo cual se utilizó la herramienta *planning poker*.

En un principio se estimó en días, lo cual dió como resultado una estimación de 66 días, donde cada día representaba 8 horas de trabajo, lo que en total sumó un total de 528 horas de trabajo. Posteriormente se estimó nuevamente, pero esta vez en horas, lo cual dió como resultado una estimación mayor, de 758 hs.

Finalmente, se realizaron estimaciones al inicio de cada sprint, lo cual en total dió una suma de 843 horas en total.

Sprint	Horas estimadas
1	56
2	80
3	219
4	210
5	278

Sin embargo, se observaron diferencias entre estas horas estimadas con respecto a las que realmente se consumieron por sprint.

Sprint	Horas reales
1	56
2	97
3	221
4	343
5	312

En comparación, se observa que, excepto en el primer sprint que no hubo una estimación previa por lo que asumimos el mismo valor, en todos los sprints los valores de las horas estimadas fueron menores que las horas realmente invertidas.

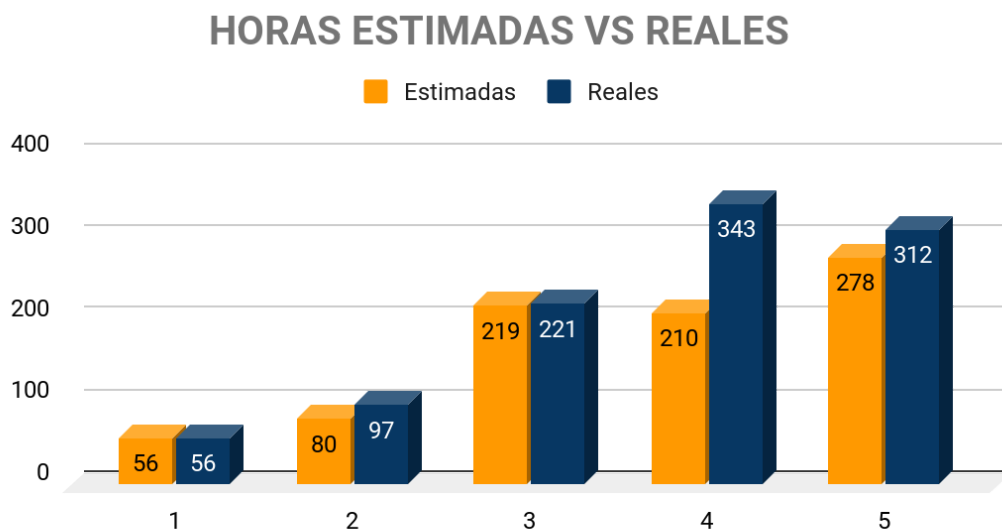


Figura 4. Gráfico de comparación entre horas estimadas vs reales.

Entonces, la suma total real de tiempo de trabajo invertido grupal fue **1029 hs**, lo cual representa un incremento aproximadamente del **22%** de la suma de las horas estimadas en cada sprint, un **35%** de la re-estimación inicial y un **95% aprox.** de la estimación inicial en días(pasado a horas).

Velocidad del Equipo

Se calcularon dos tipos de velocidades, la velocidad por sprint y la velocidad del proyecto. En ambos casos se tiene una velocidad estimada(basada en las estimaciones de tiempo) y una real(basada en los datos reales).

La velocidad por sprint la sacamos con el siguiente cálculo:

$$\frac{\text{total de puntos de historia del sprint}}{\text{cantidad de integrantes del equipo (6)}}$$

Sprint	Velocidad estimada	Velocidad real
1	9.33	9.33
2	13.33	16.17
3	36.50	36.83
4	35.00	57.17
5	46.33	52.00

Como se puede observar, al igual que con las horas de trabajo, velocidad real es superior que la estimada en todos los sprints, esto se debe a que en el mismo tiempo de trabajo se realizaron más cosas, el que cuenta con más diferencia es el sprint 4, que estimado daba una velocidad 35 y en la real aumentó un 63,35%.

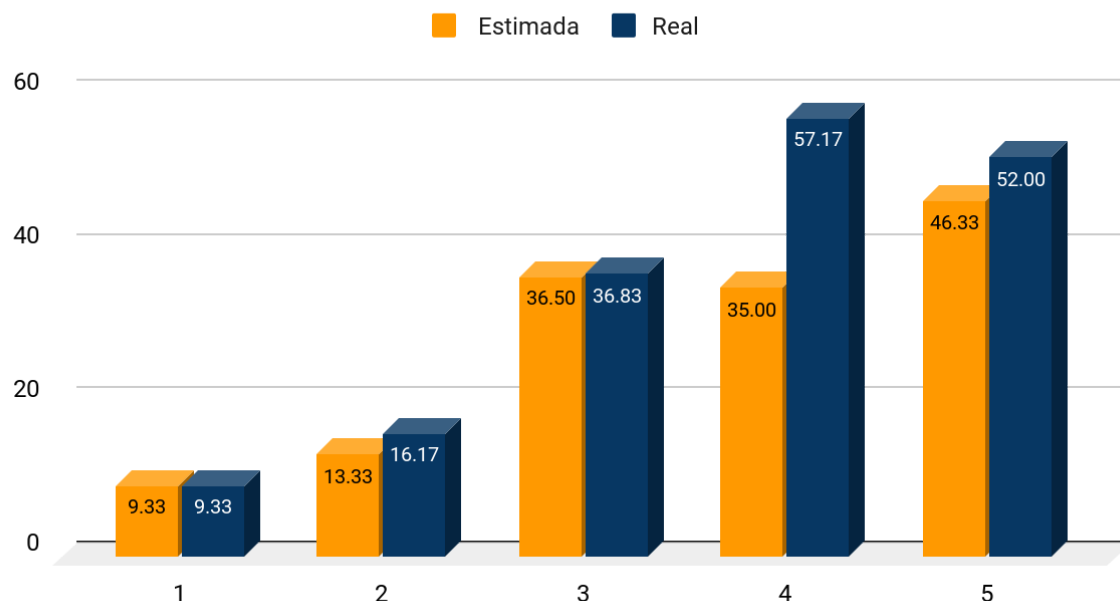


Figura 5. Gráfico de comparación de velocidades estimadas y reales.

La velocidad del proyecto la sacamos con el siguiente cálculo:

$$(\text{total de puntos de historia del proyecto})/(\text{cantidad de sprints})$$

Velocity Proyecto Estimado	Velocity Proyecto Real
168.6	205.8

La diferencia de la velocidad del proyecto general real es un 22,5% superior que la estimada. Por lo que se asume que en la realidad tuvimos una velocidad superior a la que se había estimado inicialmente.

Riesgos

Al inicio del proyecto identificamos distintos tipos de riesgos que podríamos llegar a afrontar durante el desarrollo del mismo. A lo largo de los sprints la matriz de riesgos se fue actualizando dependiendo, ya que a medida que va avanzado el proyecto los riesgos van cambiando.

Basándose en la tabla inicial, para el final del proyecto, además de estar cerrados, muchos riesgos cambiaron su exposición. Como por ejemplo:

Integración con módulos restantes.	2	3	6	Comunicación entre equipos ágiles	Pruebas de integración para evitar problemas en el negocio	Esteban Merlos
------------------------------------	---	---	---	-----------------------------------	--	----------------

Bajo su exposición ya que en el punto del quinto sprint, las integraciones ya están definidas y no hay cambios con respecto a eso en el sistema, además se realizaron pruebas de integración con el fin de no tener inconvenientes al momento del uso.

Por otro lado, hay algunos riesgos que se cumplieron, entre ellos se encuentran:

Ausencia temporal de uno o varios integrantes por problemas personales (salud, eventos externos, etc)	1	2	2	Comunicación frecuente entre miembros del equipo, charlar problemas de los integrantes.	Reasignación de tareas	Nicolas Díaz Velazquez
---	---	---	---	---	------------------------	------------------------

Ya que hubo algunas ocasiones en las que algún miembro se ha enfermado, o ha tenido que rendir exámenes/parciales de otras materias.

También, algunos riesgos se fueron cerrando ya que, el que suceda, no se consideraría un riesgo. Como por ejemplo,

Tiempo dedicado a la capacitación en nuevas tecnologías	1	2	2	Planificación de tecnologías nuevas a aprender	Ayudar entre miembros del equipo	Nicolas Díaz Velazquez
---	---	---	---	--	----------------------------------	------------------------

Ya que en este punto del proyecto no hubo más implementaciones de nuevas tecnologías que tengamos que capacitarnos.

Nombre	Probabilidad	Severidad	Exposición	Plan de mitigación	Plan de contingencia	Responsable
Retrasos durante la implementación de las funciones debido a inexperiencia	3	3	9	Investigación previa al proyecto	Comunicación con Product Owner	Ezequiel Ramallo
Integración con módulos restantes.	3	3	9	Comunicación entre equipos ágiles	Pruebas de integración para evitar problemas en el negocio	Esteban Merlos
Tiempo dedicado a la capacitación en nuevas tecnologías	3	2	6	Planificación de tecnologías nuevas a aprender	Ayudar entre miembros del equipo	Nicolas Díaz Velazquez
Bugs o trabas en la implementación	2	2	4	Pruebas unitarias	Debuggear la funcionalidad implicada	Esteban Merlos
Disponibilidad horaria de los integrantes(Estudio y trabajo)	3	1	3	Comunicación en el equipo	Asignación de tareas en el día(Dailies)	Nicolas Díaz Velazquez

Figura 6. Matriz de riesgos inicial completa en el [anexo](#).

La matriz de riesgos final completa se encuentra en el [anexo](#).

Manejo de bugs

Cada incidencia fue analizada, agregada al Backlog, implementada y validada.

Administración de cambios

Se identificaron durante el proyecto distintos tipos de fuentes de cambio

- Cambios de Tecnología:
 - De JS Reports a MetaBase
 - API del Clima
- Integración con los módulos del sistema:
 - Adaptar las tablas de la base de datos
 - Refinamiento continuo de los servicios brindados a los demás módulos.
 - Eliminación del ABM de aeropuertos

- Cambios/ Correcciones pedidas por los PO.
 - Envío de notificaciones por Mail
 - Historial de estados

Cada cambio fue estimado, agregado al Backlog, implementado y validado. Dependiendo del tamaño o impacto en el sistema se decidió en que *sprint* se implementaría el incremento, ya que cambios pequeños podían ser implementados al momento, mientras que funcionalidades nuevas que requerían más tiempo fueron postergadas al siguiente sprint a fin de no sobrecargar la planificación del trabajo actual.

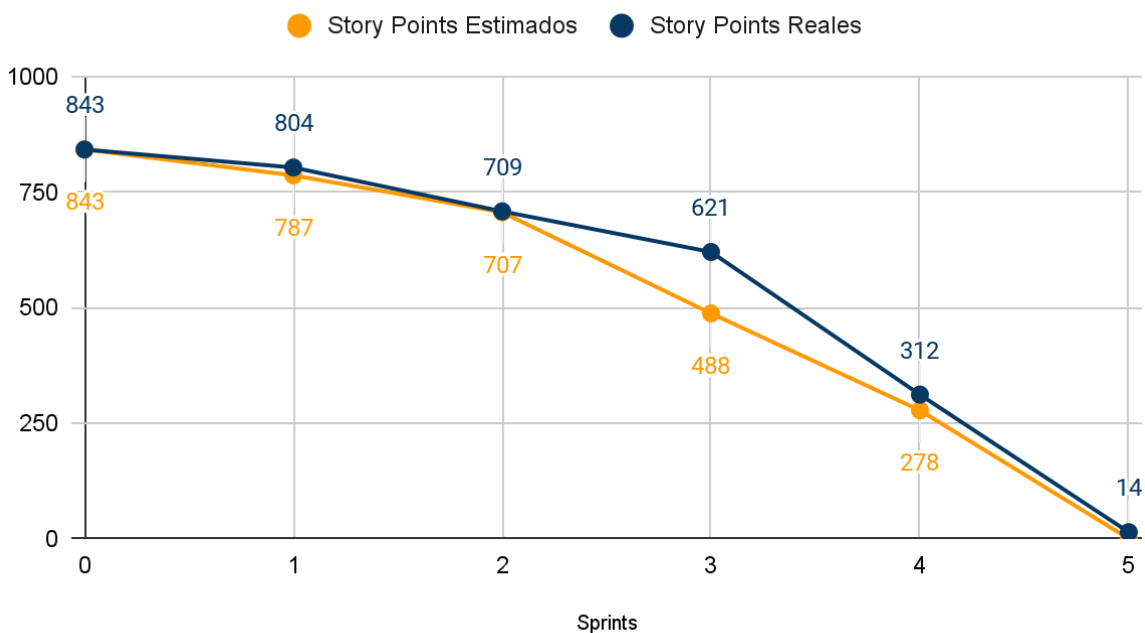
Indicadores

Burndown Chart

Decidimos realizar el gráfico de burndown chart de story points, ya que mide el proceso del proyecto pasados los sprints.

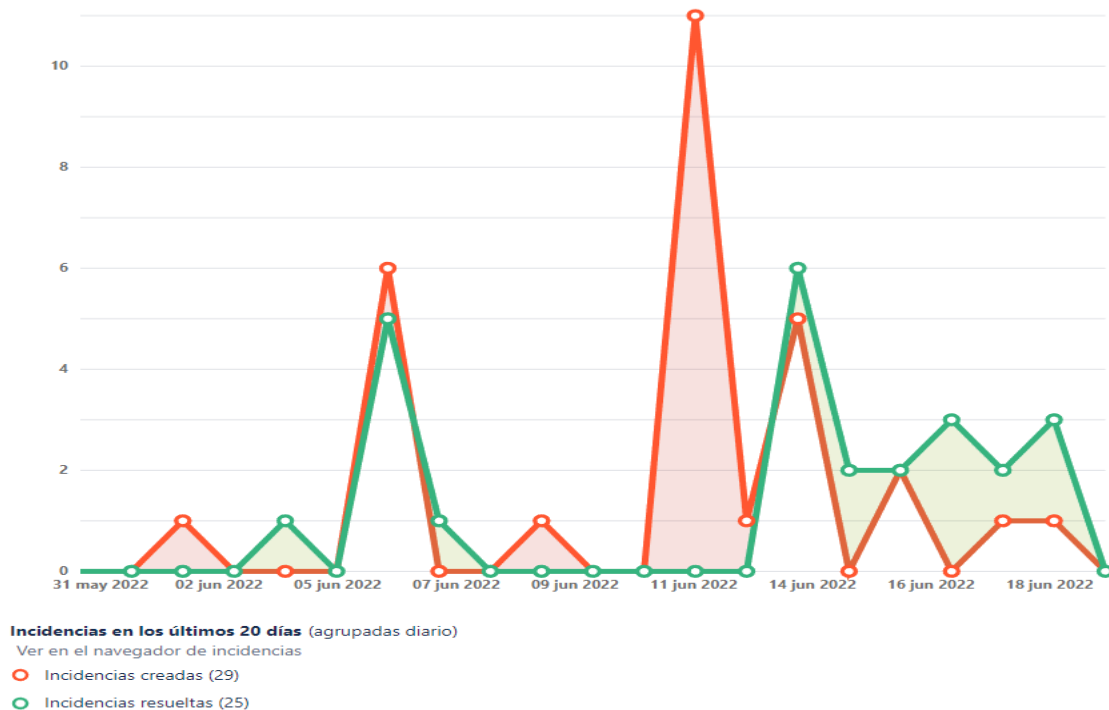
Como con las estimaciones hicimos el burndown con los puntos de historia iniciales y con los puntos de historias reales.

Podemos observar gracias al siguiente gráfico el proceso estimado del real, cómo se agregaron historias en casi todos los sprints por lo que la línea de SP reales siempre es superior.



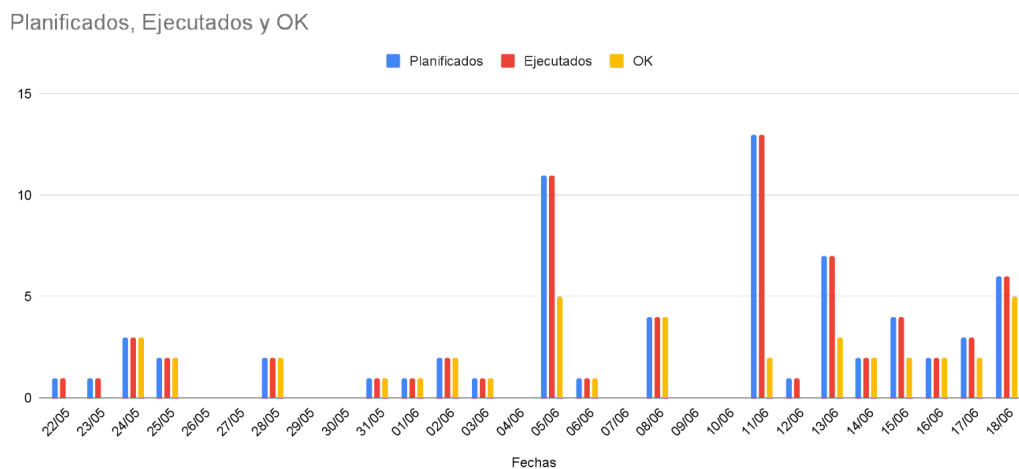
Indicador de Evolución

Este indicador nos muestra el momento en el que el proyecto se estabiliza. Esto quiere decir, el momento en el que el proyecto tiene más bugs cerrados que abiertos. Este momento se puede ver a partir del día 14 de Junio, donde se ve como la línea verde (bugs cerrados) se mantiene por encima de la línea roja (bugs abiertos).



Cobertura de la Prueba

Este gráfico, nos muestra la cobertura de las pruebas a lo largo del proyecto. En ella podemos ver los tests planificados (azul), ejecutados (rojo) y los OK (amarillo).



Integraciones

Para el correcto funcionamiento del sistema con las su totalidad, se necesitaron de integraciones, estas fueron asignadas dependiendo de las funcionalidades de cada uno de los módulos.

Que brindamos

Nuestras integraciones incluyen específicamente los planes de vuelo, al tener todos los datos de los mismos con respecto a la construcción, estado e información específica de ellos. Y además toda la información específica de cada aeronave.

Planes de vuelos dependiendo el estado del mismo

Solicitantes: Grupo 2 - Operaciones en Cabina y Mantenimiento Aeronáutico y Grupo 4 - Operaciones Terrestres, carga y despacho.

Ellos requieren la información de planes de vuelo dado el estado de los mismo, a partir de eso se le envía toda la información que para el momento del estado se encuentra cargada de los planas de vuelo.

Información específica de cada aeronave.

Solicitantes: Grupo 4 - Operaciones Terrestres, carga y despacho.

Brindamos la información técnica de cada aeronave, la matrícula, kilómetros recorridos y capacidad de la aeronave.

Historial de planes de vuelo.

Solicitantes: Grupo 1 - Operaciones Terrestres, carga y despacho

Operaciones terrestres realiza un reporte del historial de planes de vuelo, por lo tanto se debe brindarles el acceso a la información de todos ellos.

Vuelos por tripulación.

Solicitantes: Grupo 2 - Operaciones en Cabina y Mantenimiento Aeronáutico

Brindamos todos los vuelos en curso de un tripulante específico.

Vuelos por modelo de avión.

Solicitantes: Grupo 2 - Operaciones en Cabina y Mantenimiento Aeronáutico

Otorgamos información de la cantidad de veces que transita una misma ruta un modelo de aeronave.

Historial de Usuarios.

Solicitantes: Grupo 2 - Operaciones en Cabina y Mantenimiento Aeronáutico

Suministramos los usuarios registrados en nuestro sistema a fin de poner en común una misma información para ejecutar el login.

Carga destino de un vuelo

Solicitamos a: Grupo 1 - Operaciones Terrestres, carga y despacho

Informamos al grupo 1 el aterrizaje de un vuelo para que gestionen la carga en el destino.

Que requerimos

Para el correcto funcionamiento de nuestro sistema, necesitamos muchos datos que pertenecen a otros módulos:

Rutas:

Solicitamos a: Grupo 4 - Operaciones Terrestres, carga y despacho.

Lo requerimos para recibir las rutas reales y con ello obtenemos los vuelos planificados, gracias a esta información podemos comenzar con la planificación de los vuelos.

Carga origen de un vuelo

Solicitamos a: Grupo 1 - Operaciones Terrestres, carga y despacho

Lo requerimos para una vez aterrizado cargar la información de la carga destino del mismo. Es uno de los datos que tenemos que persistir del vuelo.

Aeropuertos

Solicitamos a: Grupo 4 - Operaciones Terrestres, carga y despacho

Los requerimos para en principio saber tener la ubicación real de los tripulantes y aeronaves. Y además utilizamos esta información para mostrar la ubicación de los aeropuertos en el mapa.

Ruta Alternativa / Ruta Real

Solicitamos a: Grupo 4 - Operaciones Terrestres, carga y despacho

Lo requerimos porque un vuelo puede ser reprogramado entonces debemos pedir una ruta alternativa para asignarla a ese vuelo. También precisamos esta información cuando durante un vuelo se produjo algún inconveniente y se cambió el destino, nosotros debemos persistir esa información.

Información técnica de modelo de aeronave

Solicitamos a: Grupo 4 - Operaciones Terrestres, carga y despacho

Lo requerimos porque nosotros tenemos el maestro de cada aeronave, donde guardamos los datos específicos de los modelos. Cómo saber cuánto combustible consumen.

Control de cabina

Solicitamos a: Grupo 2 - Operaciones en Cabina y Mantenimiento Aeronáutico

Antes de despegar requerimos los controles de cabina aprobados para poder cambiar el estado de un plan de vuelo a despegado.

Emergencias en un vuelo

Solicitamos a: Grupo 2 - Operaciones en Cabina y Mantenimiento Aeronáutico

Lo requerimos porque una vez que aterriza el avión, también se deben cargar todos los datos del vuelo uno de ellos es si hubo emergencias DURANTE el mismo.

Mantenimiento de aeronaves

Solicitamos a: Grupo 2 - Operaciones en Cabina y Mantenimiento Aeronáutico

Lo requerimos porque al asignar una aeronave a un vuelo precisamos conocer si se encuentran en condiciones de despegar.

Insumos por vuelo

Solicitamos a: Grupo 2 - Operaciones en Cabina y Mantenimiento Aeronáutico

Lo requerimos para cargar los insumos al vuelo. Además requerimos la cantidad de insumos final de un vuelo.

Integraciones con APIs externas

Además de integraciones con los módulos restantes del sistema, para nuestro subsistema precisamos APIs externas que tienen un funcionamiento específico para cumplir con algunas funcionalidades que nuestro sistema posee.

Open Meteo

Lo requerimos para calcular el clima en el aeropuerto de llegada y así poder controlar las condiciones de un vuelo.

Sistema de mapas leaflet

Lo requerimos para la visualización en mapa de los aeropuertos de salida y llegada.

Retrospectiva

Inconvenientes surgidos y resoluciones

Como primera experiencia realizando este tipo de proyectos no vamos a negar que tuvimos que enfrentar algunos inconvenientes, vamos a nombrar algunos de ellos con su explicación y su resolución.

No respetamos al pie de la letra las ceremonias de Scrum.

En un principio nos costaba mucho seguir las dailies como se deben hacer, ya que realizamos la llamada diaria pero dentro de ella nos dispersamos del tema. Lo que sí, se terminaban tocando todos los temas que se necesitan para realizar bien la ceremonia, pero esta, en vez de durar 15 min duraba 4 horas.

A lo largo de cada sprint supimos manejar esta situación un poco más disminuyendo el tiempo invertido en estas en 1 o 2 horas. Igualmente este tipo de “dailies” nos sirvieron ya que era una puesta grupal que incluyó ayudas y puntos de vista de todo el equipo.

No separar apropiadamente las historias de usuario de gran tamaño.

Cuando armamos el backlog creamos historias de usuario que guardaban mucho contenido, eran muy largas y se sobrecargó a miembros del equipo. Este error lo cometimos varias veces ya que no se estimó de manera correcta la complejidad de ciertas funcionalidades.

Esta situación la pudimos solucionar en los últimos sprint desglosando funcionalidades en más historias.

No tener al día los avances de las User stories.

Nos costó priorizar la gestión del proyecto antes que la programación del producto, por esta razón durante los primeros sprints quizás habíamos terminado las funcionalidades requeridas pero nunca las cerramos en JIRA. Esto nos perjudicaba al momento de hacer el burndown chart de los sprint que figuraba como si hubiésemos hecho todo el trabajo a último momento cuando en muchas ocasiones no era así.

Pudimos manejar esta situación con recordatorios grupales casi diarios de que actualicemos los cambios realizados en el backlog.

Lecciones aprendidas

Gracias al trayecto de este proyecto pudimos aprender distintos tipos de cosas, formas en las que nos manejamos, decisiones que tomamos, dentro de algunas a destacar nos encontramos con dos categorías:

¿Qué volveríamos a hacer?

Realizar llamadas entre miembros del equipo para afrontar funcionalidades (PAIR PROGRAMING), este tipo método para afrontar la programación nos ayudó al mejor entendimiento de las funcionalidades, quizás si alguno mientras desarrollaba se estancaba podía pedir ayuda en el momento para resolver el conflicto en el que se encontraba, esto también ayudó a evitar la generación de estrés innecesario. No era estrictamente necesario que se trabaje en lo mismo, pero tener un soporte en el momento del conflicto era sin duda una salvedad.

Elección de las tecnologías de desarrollo, la elección de las tecnologías de desarrollo fue pensada específicamente para que tanto desarrolladores del backend como del frontend puedan comprender todo lo que se realizaba ya que estuvimos trabajando en el mismo lenguaje. Esto ayudó en momentos a que cualquiera de los desarrolladores puedan realizar cambios o reparar bugs que se encontraban y poder tener más rapidez en la realización de estos.

Constante comunicación entre los miembros del equipo, la comunicación es esencial en un equipo, nosotros contamos con un grupo de WhatsApp, donde diariamente nos comunicamos, y tenemos una buena relación. Gracias a esto pudimos organizarnos correctamente en momentos en los que contábamos con ausencias del equipo por diversas razones, ya que estas eran avisadas con antelación.

¿Qué NO volveríamos a hacer?

Extender de manera exagerada las llamadas entre todos los miembros del equipo. Esto es un problema

que identificamos a la hora de hacer las dailies, como estaba todo el equipo y estas se extendían se perdía mucho tiempo de trabajo individual.

Subestimar la elección de la tecnología para los reportes. Al momento de elegir la herramienta de los reportes revisamos muy por arriba y no fuimos precavidos de ver si la herramienta era compatible con lo que queríamos hacer nosotros y efectivamente la que elegimos al inicio no lo era. Por esta razón a último momento tuvimos que cambiar a Metabase que pudimos adaptarla a nuestras preferencias.

Mala gestión de tiempos ociosos entre los miembros de distintas secciones del equipo. Esto tiene que ver más que nada con la mala división de tareas, hubo sprints en que algunos miembros estuvieron más cargados de tareas que otros. Gracias a esto empezamos a realizar una división más exigente de las tareas.

Conclusión

Conclusiones del producto

Una vez finalizado el desarrollo del sistema se realizó un análisis del mismo, teniendo en cuenta las funcionalidades ofrecidas, el diseño de las interfaces, y la usabilidad de estas.

Dentro de los aspectos observados, se destacan:

- Cumplimiento de todas las funcionalidades requeridas inicialmente,
- Look and Feel agradable a la vista,
- Interfaz intuitiva para el usuario.

Conclusiones del proyecto

El desarrollo del proyecto inició con una gran incertidumbre debido a la complejidad del mismo ya que ninguno de los integrantes del equipo había realizado uno similar, en especial, con respecto a las integraciones con otros módulos. Se cometieron algunos errores en las estimaciones iniciales debido a la falta de experiencia con proyectos de esta dimensión.

Otro factor que era inicialmente desconocido para el equipo fue la gestión específica de un proyecto; existieron dificultades en la adaptación a la metodología y, a su vez, en la priorización de la gestión en vez de la implementación. Esto se observó por ejemplo, al momento de aplicar las ceremonias propias de la metodología *Scrum*.

En general, se puede afirmar que la experiencia obtenida es para destacar, como así también el conocimiento adquirido relacionado tanto a la gestión de un proyecto como a las nuevas herramientas de programación. Estos conocimientos permitieron un ágil despliegue a lo largo del proyecto.

Finalmente, queremos expresar la satisfacción del equipo con respecto a la gestión final del proyecto y el producto obtenido. Podemos afirmar que vemos una gran evolución con respecto al plan de proyecto inicial en comparación con el final y, aunque se hayan tenido errores, estos nos sirvieron para tener un mejor aprendizaje y de esta forma poder utilizar este tipo de gestión en proyectos futuros.

Repositorios

Durante todo el proyecto utilizamos el repositorio que fijamos al principio, Gitlab. En la rama main está todo el trabajo principal con todas las actualizaciones, las otras ramas existentes son trabajos de los distintos miembros del equipo que luego se actualizaban en la rama principal.

Link de Gitlab: [Repositorio Icarus](#)