

Memoria



Proyecto Colibrí

Laboratorio de desarrollo de software

Grupo Paire

Ariel Machini

Cinthia Lima

Julio del 2020



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

La memoria del proyecto describe todo el proceso de desarrollo del sistema Colibrí: desde su comienzo hasta su culminación. Este documento no sólo se centra en la descripción de las actividades técnicas, sino también en como atravesó y experimento el equipo de trabajo dicho proceso de desarrollo.



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

Índice

Introducción.....	4
Objetivos.....	5
Objetivo general.....	5
Metodología de desarrollo.....	5
Organización general del trabajo.....	6
Comunicación del equipo.....	7
Gestión del proyecto.....	7
Estimación y planificación del proyecto.....	8
Gestión de riesgos.....	12
Validación y verificación del software.....	23
Herramientas y tecnologías.....	24
Gestión del proyecto.....	24
Clases, lenguajes y frameworks.....	25
Herramientas para el desarrollo.....	26
Modelado.....	27
Otros.....	27
Desarrollo del proyecto.....	28
Primera fase: Fase de inicio.....	28
Segunda fase: Fase de elaboración.....	29
Tercera fase: Fase de construcción.....	30
Cuarta fase: Fase de transición.....	32
Conclusión.....	32



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

Memoria

Introducción

En esta memoria se detallará el proceso mediante el cual se desarrolló el producto de software Colibrí, solicitado para la aprobación de la materia Laboratorio de Desarrollo de Software de la carrera Licenciatura en sistemas. La descripción del proceso se abordará desde una perspectiva en la cual se deje en evidencia cómo el equipo de desarrollo logró llevar a cabo con éxito las tareas necesarias para terminar el sistema. Más allá de la documentación técnica, la cual es inherente al desarrollo de este proyecto, lo que quiere mostrar esta memoria es como se vivió el proceso de desarrollo, qué problemas se encontraron y cómo se solucionaron, qué estrategias se eligieron para solucionarlos y qué aprendizaje o aumento de habilidades se obtuvo gracias al desarrollo de este producto de software.

Colibrí consta principalmente de una aplicación web para la creación y gestión de formularios personalizables, con su funcionalidad de contraparte, que es permitir el rellenado de tales formularios desde la misma aplicación web o desde una aplicación móvil. Una característica a destacar de la aplicación web es que esta trabaja con un subsistema llamado UARGFlow, el cual permite distinguir entre diferentes tipos de usuarios y gracias a ello cuando se crean formularios se puede determinar a qué tipos de usuarios van dirigidos dichos formularios. Este producto de software fue solicitado para su puesta en funcionamiento en la Unidad Académica de Rio Gallegos (UARG) de la Universidad Nacional de la Patagonia Austral (UNPA), ya que en la actualidad los formularios (o actividades que se podrían hacer de manera más eficiente mediante uno) de varias áreas de la universidad no están digitalizados. Los formularios digitalizados frente a los que están en papel hacen que las tareas de distribución y recopilación de respuestas sea más fácil y cómoda.

Para llevar a cabo el desarrollo del sistema, se uso un marco de trabajo que permitiera organizar, planificar y controlar todas las actividades necesarias para cumplir con el objetivo. Se utilizó la metodología de desarrollo [PSI](#), la cual brinda el marco necesario para el desarrollo de las actividades que incluye una colección de plantillas para documentar el proceso de desarrollo de software.

En las secciones siguientes se especificarán: los objetivos del proyecto; cómo se organizó el equipo para poder llevar a cabo las actividades del



Grupo Paire

Ariel Machini y Cinthia Lima

proceso de desarrollo; como se llevó a cabo en la gestión del proyecto; las herramientas y tecnologías utilizadas; el desarrollo del proyecto dividido en 4 fases y por último las conclusiones finales.

Objetivos

Objetivo general

El objetivo general de este desarrollo es brindar una solución tecnológica funcional y de calidad, siguiendo un proceso organizado, controlado y documentado de todas las actividades llevadas a cabo necesarias mediante el marco de trabajo PSI para la materia Laboratorio de desarrollo de software.

Objetivos específicos

- Desarrollar una aplicación web que englobe todas las funcionalidades solicitadas por el cliente.
- Desarrollar una aplicación móvil para el rellenado de formularios de acceso público.
- Llevar a cabo el desarrollo dentro de los tiempos impuestos por los clientes.
- Entregar las aplicaciones con sus respectivos manuales de usuario y de instalación.

Metodología de desarrollo

Como se mencionó anteriormente la metodología de trabajo elegida fue [PSI](#). Esta metodología de desarrollo esta basada en el Proceso Unificado para el desarrollo de software, la cual se caracteriza por estar dirigida por Casos de uso, centrado en la arquitectura y por ser iterativo e incremental. Esta metodología fue desarrollada como Proyecto Final de la carrera Licenciatura en sistemas en la UARG. Este marco de trabajo brinda una pauta general de cómo llevar a cabo el proceso de desarrollo, qué actividades realizar, cómo realizarlas, cuando y porque. Además, ofrece un conjunto de plantillas explicadas, acompañadas de ejemplos para la generación de los artefactos de documentación inherentes a cada fase de desarrollo. Las plantillas que ofrece PSI son de suma importancia ya que



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

son una guía clave para realizar una documentación adecuada del proyecto y más en un desarrollo de este índole debido a que es una primera experiencia en el proceso del desarrollo de software. La documentación es fundamental para las etapas de mantenimiento del sistema y también para analizar el proceso de desarrollo y así poder mejorar de manera continua en los procesos realizados para futuros proyectos.

El proceso que propone PSI se divide en 4 fases: inicio, elaboración, construcción y transición. En cada fase se desarrollan un conjunto n de iteraciones. Las iteraciones dan como resultado un incremento del producto desarrollado, ese incremento puede ser un artefacto o implementación que añade o mejora las funcionalidad del sistema en desarrollo. A su vez cada una de las iteraciones se realizan actividades relacionadas en mayor o menor medida con las diferentes disciplinas según la fase en la que se encuentre la iteración. Las principales disciplinas involucradas son: modelado del negocio, gestión del proyecto, requerimientos, análisis y diseño, implementación, pruebas e implantación.

Organización general del trabajo

Durante el primer mes del proceso de desarrollo el equipo de trabajo estaba conformado por 3 miembros, pero luego por motivos de fuerza mayor, el equipos se redujo a 2 y así se mantuvo durante todo el desarrollo. Para llevar a cabo las diferentes actividades del proyecto se realizaron 15 iteraciones en las cuales se definían distintas actividades y se asignaban a alguno de los miembros del equipo. Estas iteraciones se documentaron en planes de iteración, los cuales van acompañado de una plan más general denominado “Plan de proyecto”.

Cada iteración tenía una fecha de cierre, en la cual el equipo de trabajo verificaba sin falta que actividades se pudieron cumplir, cuales no y porque. Esto quedaba documentado en el plan de iteración junto con una conclusión de la iteración. La mantención de los planes de iteración durante todo el desarrollo fue de gran ayuda para organizar y controlar el trabajo realizado.

Comunicación del equipo

La buena comunicación entre el equipo de trabajo es clave para poder organizar el trabajo. El equipo de trabajo convive en un mismo lugar físico, por ende, la comunicación en la mayoría de los casos fue en persona y en el



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

momento que fuese necesaria. Esto facilitó el tratamiento de los diferentes aspectos que se tenían que discutir acerca del proyecto, ya que fueron tratados de manera rápida, efectiva y en el momento en el que surgían. Es necesario mencionar que también existieron un par de iteraciones que realizadas de manera virtual ya que, al momento de realizarlas, el equipo de desarrollo no se encontraba conviviendo. Para realizar dichas iteraciones la comunicación se llevaba a cabo mediante el servicio de mensajería WhatsApp.

Calidad y gestión del proyecto

La gestión de proyectos de software es una actividad esencial en el desarrollo de software que debe estar presente antes de iniciar cualquier actividad técnica y continuar durante todo el desarrollo. Una falta de gestión puede llevar al fracaso a cualquier proyecto de software, es por eso que es muy importante tener una gestión del proyecto desde que este inicia. Uno de los objetivos principal de la ingeniería del software es generar software de calidad, y por ende debe existir una gestión de calidad. Existen varias actividades a realizar para asegurar la calidad del software, y entre ellas se encuentra la gestión del proyecto en términos de planificación y estimación del proyecto, entre otras actividades.

Durante el desarrollo del proyecto se estuvieron realizando las actividades para el aseguramiento de calidad, pero no estaban explícitas en un Plan de calidad, por ende en una de las iteraciones finales decidimos explicitar en un Plan de calidad qué actividades de aseguramiento y control se iban a realizar para producir un sistema de alta calidad. Reconocemos que el Plan de calidad se tendría que haber realizado con anterioridad, pero en el principio del desarrollo no fue propuesto como un entregable obligatorio, por eso lo realizamos más adelante cuando tuvimos tiempo para redactarlo. Las actividades especificadas en dicho plan fueron las siguientes:

- Estimación y planificación del proyecto
- Gestión de riesgos
- Validación y verificación del software

Estimación y planificación del proyecto

Las actividades de estimación y planificación son dos de las actividades más importantes y que se deben hacer antes de comenzar con cualquier



otra actividad y deben repetirse continuamente durante todo el desarrollo, ya que se ven influenciadas por la evolución del producto de software a desarrollar.

La estimación del tamaño y esfuerzo necesario para llevar a cabo un proyecto de software es una de las primeras actividades a realizar y para ello se debe estudiar el problema a resolver y definir su alcance, así como también las características del entorno y equipo de trabajo. Para poder recopilar toda la información para realizar la estimación se hizo un estudio de factibilidad, un modelado de negocios, una obtención y modelado, mediante casos de uso, de los requerimientos.

La estimación en este proyecto no pudo ser realizada inmediatamente después de haber recopilado la información necesaria ya que no contábamos con las herramientas y práctica necesaria para llevarla a cabo en ese momento. Además, si bien se había hecho un estudio de factibilidad y se conocían las características del sistema a desarrollar, existían requerimientos que se iban definiendo o cambiando durante la cursada de la materia en la cual se despliega el desarrollo de este sistema. Las 3 primeras iteraciones se desarrollaron en el marco de la cursada de la materia sin tener una previa estimación. La primera estimación fue realizada recién al terminar la 3ra iteración, para dicho momento el equipo tenía una visión muy clara del sistema a desarrollar, conociendo que actividades quedaban pendiente y cuales habían sido completas, y con esa información fue que se trabajó en la primera estimación.

El método elegido para realizar la estimación fue mediante Puntos de caso de uso. Durante todo el proceso de desarrollo se realizaron 3 estimaciones, y a continuación se describirá los resultados de cada estimación y en que condiciones se realizaron:

- Estimación 1 (12 meses trabajando 12 horas por semana): esta primera estimación se hizo el 24/11/2017. Y el estado del proyecto estaba avanzado ya que hasta ese entonces se habían realizado 3 iteraciones con muchas actividades definidas.
- Estimación 2 (3,5 meses trabajando 21 horas por semana): esta segunda estimación se realizó el 22/03/2020, y se hizo cuando se retomó el proyecto luego de trabajar en algunas iteraciones durante el 2019. Desde la última estimación hasta esta ya se realizaron 4 iteraciones (durante el año 2019), en las cuales se logró avanzar considerablemente en el desarrollo y también en los conocimientos y capacidades del equipo de desarrollo, es por esto que se puede notar una gran diferencia en la duración de esta estimación en comparación



con la primera. Como se puede ver en la Figura 1, donde la línea naranja representa el tiempo de esta estimación número 2 en comparación con la 1.

- Estimación 3 (1,5 meses, trabajando 14 horas por semana): esta estimación se hizo el 04/05/2020 y en esa fecha ya se habían llevado a cabo 4 iteraciones desde la última estimación. El proyecto se encontraba en un estado bastante avanzado en este momento y sólo quedaban pocas tareas pendientes.

En la Figura 1 se puede apreciar la evolución de los tiempos estimados de duración del proyecto para el Grupo Paire, así como también la diferencia entre esos tiempos en contraste con los tiempos estimados para una persona. Esto nos permite decir que es notable la diferencia cuando la cantidad de personal de trabajo es mayor a 1. De igual manera, como algunos pueden concluir en que sumando personal se obtienen tiempos más bajos, hay que aclarar que eso siempre dependerá mucho de como se gestione a dicho personal. La solución no siempre es agregar más personas.

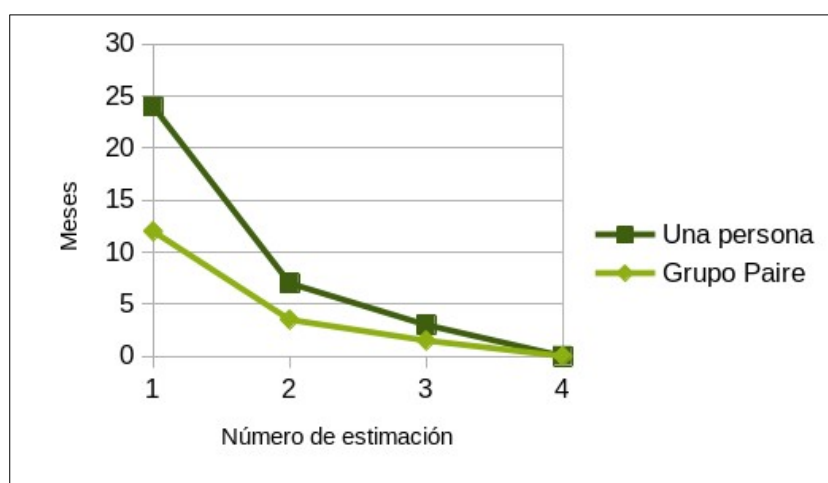


Figura 1: Estimación del proyecto

La duración real del desarrollo en comparación con las estimaciones realizadas fue mucho menor. En la Figura 2 se puede ver como todas las estimaciones son mayores a la duración real del proyecto, considerando el tiempo desde esa estimación hasta el fin del desarrollo. Por otro lado, la diferencia entre la estimación y la duración real disminuye cada vez que se realiza una nueva estimación. La disminución sucede debido a que en cada estimación se realiza con información más exacta y además va acompañado del avance en el estado del proyecto.



Grupo Paire

Ariel Machini y Cinthia Lima

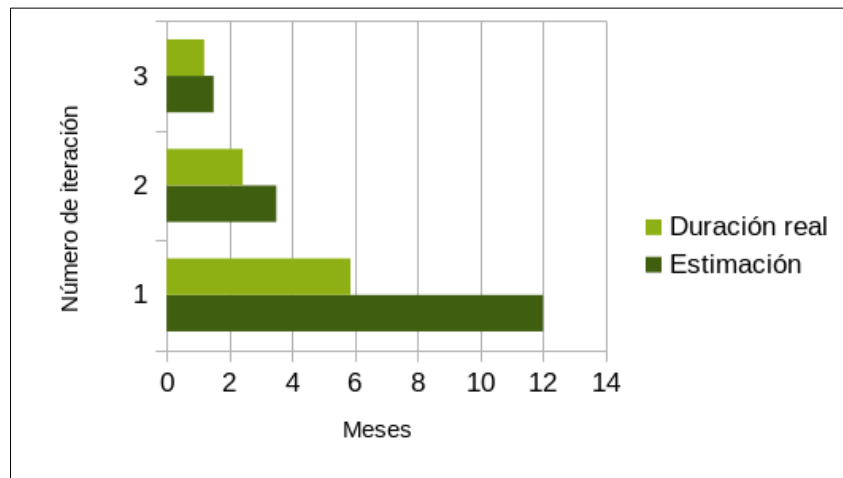





Figura 2: Comparación de la estimación con la duración real





La planificación forma parte de la gestión efectiva del proyecto. Se planificaron de manera detallada las actividades de ingeniería que se iban a llevar a cabo en cada iteración y se supervisó su progreso para verificar su cumplimiento. Para ello se elaboraron planes por cada iteración, en los que se especificó los recursos necesarios (humanos y materiales), se dividió y asignó el trabajo (actividades) y se creó definió un calendario de trabajo (tiempos). Estos planes de iteración a su vez, alimentan a un plan de proyecto mucho más general, en el cual se puede ver el avance y estado del proyecto en un momento dado. Se planificaron 15 iteraciones, de las cuales 3 fueron realizadas en el año 2017, 4 en el 2019, y 8 en el año 2020. En año 2017 nos encontrábamos trabajando en el proyecto dentro de la materia Laboratorio de desarrollo de software, manteníamos iteraciones cargadas de actividades y al mismo tiempo nos estábamos familiarizando con la puesta en práctica de la metodología de desarrollo PSI y en términos más generales con el proceso de desarrollo de software en general. Las cantidad de iteraciones, la duración y a que fase pertenecen cada una se puede ver en la Tabla 1.

Tabla 1: Iteraciones en el año 2017

Iteración	2017			
	Agosto	Septiembre	Octubre	Noviembre
F1 I1				
F2 I1				
F3 I1				
Días	18		54	10

Luego de terminar la cursada en el año 2017, recién retomamos el proyecto en el año 2019 porque en el 2018 nos encontrábamos ocupados cursando varias materias, aprobando finales pendientes, y realizando otras actividades académicas lo cual no nos permitía trabajar en el proyecto. En enero del 2019 retomamos el desarrollo del proyecto, trabajamos casi hasta mitad del año con una interrupción en marzo y abril porque reanudábamos el año académico, la otra mitad del año también nos encontrábamos adelantando materias y otras actividades académicas que no nos dejaban tiempo para continuar trabajando. El desarrollo de las iteraciones en el 2019 se puede ver en la Tabla 2.

Tabla 2: Iteraciones en el año 2019

Iteración	2019						
	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio
F3 I2							
F3 I3							
F3 I4							
F3 I5							
Días	48				23	13	20

En marzo del año 2020 retomamos el proyecto en el estado que se encontraba en Julio del 2019. En este nuevo año trabajamos desde marzo hasta la agosto, casi sin interrupciones a excepción de unos días de mayo en los que realizamos un Seminario y todo el mes de junio, en el que nos encontrábamos cerrando el cuatrimestre de cursada y trabajando en un proyecto para aprobar el Seminario cursado en mayo. En la Tabla 3 se puede ver como se desarrollaron las iteraciones.



Tabla 3: Iteraciones en el año 2020

Iteración	2020					
	Marzo	Abril	Mayo	Junio	Julio	Agosto
F1 I6	■					
F2 I7		■				
F3 I8		■				
F3 I9			■			
F3 I10				■		
F4 I1					■	
F4 I2						■
F4 I3						■
Días	9	9 10 10	9		9 7	11

Se realizó el seguimiento de los planes, durante todo el proceso, ya que es una actividad fundamental en el proceso de la planificación porque haciendo un seguimiento se pueden detectar rápidamente los problemas de planificación, cuando aún queda tiempo para poder resolverlos.

Tanto la estimación como la planificación fueron procesos iterativos que estuvieron presentes durante el desarrollo del proyecto. Tenemos que reconocer que las estimaciones realizadas fueron pocas, y que si bien nos ayudaron conocer el tiempo estimado necesario en este proyecto, hay que tener en cuenta que no todos los proyectos son iguales, y que no hacer estimaciones con mayor frecuencia puede perjudicar la entrega del producto en tiempo y forma. Por otro lado la planificación de cada iteración siempre fue realizada, su seguimiento y duración fue evolucionando, en un principio (2017) las planificaciones fueron algo caóticas debido a las cantidad de tareas, la complejidad de estas y la poca experiencia del equipo de trabajo. Pero a medida que avanzábamos en el desarrollo planificar y realizar el seguimiento de las tareas eran cada vez actividades que se realizaban de manera natural y no generaban tantos conflictos como en el principio.

En términos de la fecha límite de entrega del proyecto definida por la regularidad de la materia no tuvimos problemas. La fecha límite de entrega del proyecto era finalizados los dos años y medio de regularidad de la materia Laboratorio de desarrollo, en nuestro caso el plazo de esa regularidad terminaba en julio de este año, pero debido al contexto de pandemia que atravesamos las regularidades se extendieron.



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

Gestión de riesgos

La gestión de riesgos involucra la estimación de los posibles riesgos a los que está expuesto el proyecto en desarrollo, también involucran las tareas de análisis, priorización y determinación de estrategias para minimizar el impacto que estos puedan llegar a tener. Los riesgos son cualquier suceso que de llegar a suceder pueden afectar negativamente a diferentes aspectos del desarrollo del proyecto que pueden afectar su planificación. La metodología PSI nos ofrece una plantilla automatizada para identificar y evaluar los riesgos, y también nos brinda una lista ordenada con los riesgos de mayor a menor prioridad la cual es calculada de acuerdo al impacto y probabilidad del riesgo, dicha lista ordenada nos permitía decidir de manera más fácil que riesgos gestionar. La plantilla nos propone clasificar los riesgos identificados en 10 diferentes categorías, pero por las características de nuestro proyecto los riesgos identificados se clasificaron en 7 categorías, y no se identificaron riesgos en las categorías denominadas: aspectos financieros, subcontratista y legal y contractual.

La gestión de los riesgos es una actividad que requiere tiempo para realizar su identificación y seguimiento continuamente. Es por esta razón que no se gestionaron todos los riesgos identificados, todos fueron considerados pero no se pudieron planear actividades de mitigación ni hacer un seguimiento de todos. La toma de decisión acerca de que riesgos iban a ser gestionados y cuales no se realizó en equipo y siempre se tenía en consideración el estado y características actuales del proyecto en ese momento.

Durante todo el proceso de desarrollo se realizaron 6 estimaciones y evaluaciones de los riesgos.

Primera estimación y evaluación de riesgos — Fase 2, iteración 1

En esta primera iteración se identificaron 17 riesgos de los cuales luego de una evaluación sólo consideraron 15. Estos riesgos se pueden apreciar en la Tabla 4 organizados por categoría y ordenados por prioridad, y la Figura 3 que muestra en un gráfico como se distribuyeron los riesgos en cada categoría.

Tabla 4: Primera estimación de riesgos

Categoría	ID Riesgo	Riesgo
Experiencia y capacidad	R05	El grupo de desarrollo no tiene experiencia en cuanto a la conducción



		de un proyecto de software
	R06	El grupo de desarrollo no tiene experiencia en el desarrollo de software
	R07	No se conoce el/los lenguaje(s) de programación a utilizar ni como trabajan.
	R09	No se tiene conocimiento ni experiencia respecto al funcionamiento del API de Google Forms.
	R10	Falta de experiencia en desarrollo de aplicaciones móviles.
Cronograma	R04	Puede que el límite de tiempo establecido no sea suficiente para desarrollar el proyecto
Definición del cliente	R02	El cliente puede proponer grandes cambios para el proyecto.
	R03	No se definieron todos los documentos a entregar junto con el producto
Compromiso del cliente	R01	El cliente no estará presente durante todo el proceso de desarrollo
Experiencia y capacidad	R08	No se tiene experiencia con el modelo de desarrollo PSI
Tecnología	R11	La adaptación del modulo Workflow a nuestra aplicación puede ser difícil.
	R15	La preparación del área de trabajo con las tecnologías a utilizar puede ser difícil
Complejidad	R12	Los entregables pueden ser muy complejos y puede llevar tiempo su creación.
Duración y tamaño	R13	Las partes del sistema que no se especificaron consumen más tiempo de lo esperado
	R14	El aprendizaje de las nuevas tecnologías puede llevar más tiempo



		del esperado.
--	--	---------------

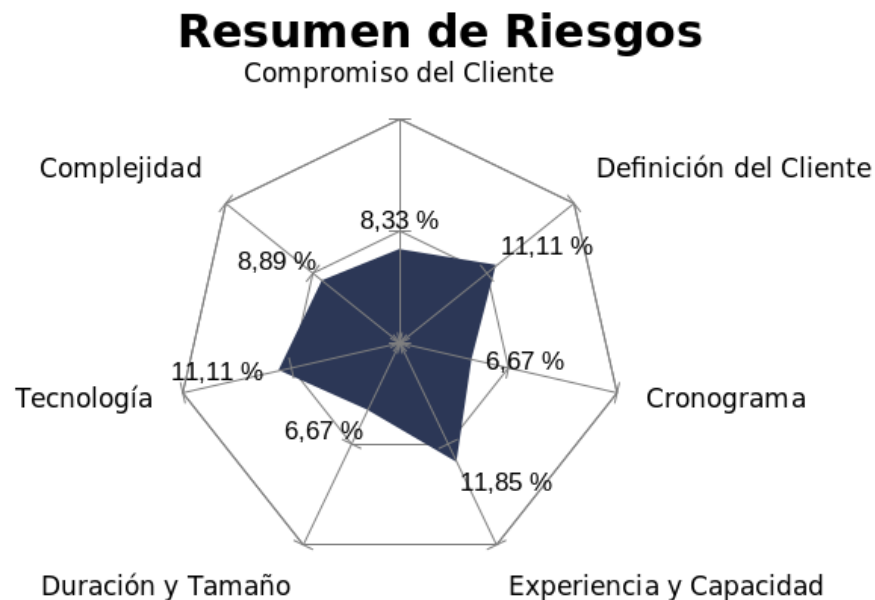


Figura 3: Distribución de los riesgos identificados en la diferentes categorías (I)

Se decidió gestionar y hacer un seguimiento de los riesgos R05, R07 Y R09, debido a que eran, según nuestro análisis los que tenían una probabilidad muy próxima de suceder e impactar en nuestro proyecto. Si, bien el riesgo R06 se encuentra como uno de los segundo en términos de prioridad, se decidió no hacer una planificación y seguimiento del mismo porque, la única manera de obtener esa experiencia que no teníamos y mitigar ese riesgo era continuar trabajando en el desarrollo del proyecto.

En la Figura 3 se puede apreciar como en esta primera estimación la mayoría de los riesgos identificados se encontraban en las categorías de Experiencia y capacidad y en Tecnología. Esta distribución de los riesgos tiene mucho sentido, ya que el equipo se encontraba iniciando el desarrollo y conociendo las tecnologías que teníamos que usar.

A continuación, en la Tabla 5, se detallará que acciones se planificaron frente a estos riesgos y su estado en la iteración 1 de la fase 2.



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

Tabla 5: Gestión de riesgos I

Riesgo	Acción
R05	<i>Estrategia de reducción:</i> se definió un marco de gestión para dirigir el proyecto. Para ello se verificaba continuamente que las tareas de planificación de iteración del proyecto siempre fueran realizadas y las actividades planificadas fueron ejecutadas.
R07	<i>Estrategia de reducción:</i> se definieron 3 tareas: determinar que lenguajes se van a utilizar; estudiar los lenguajes a medida que se desarrolla el proyecto; desarrollo de pequeños programas de prueba con los lenguajes estudiados.
R09	<i>Estrategia de eliminación:</i> se definió buscar información a cerca de la existencia o no de esa posible API, preguntando en foros de desarrolladores y investigando al respecto.

Segunda estimación y evaluación de riesgos — Fase 2, iteración 1

En esta segunda evaluación de riesgos, de los 15 riesgos identificados en la primera estimación se descartaron 8, ya sea porque se aplicó alguna estrategia para reducirlos o eliminarlo o porque cambiaron los parámetros de análisis del equipo de desarrollo. También aparecieron nuevos riesgos y cambiaron algunas prioridades. En la Tabla 6 se pueden ver el nuevo estado del catálogo de riesgos en esta nueva evaluación.

Tabla 6: Segunda estimación de riesgos

Categoría	ID Riesgo	Riesgo
Cronograma	R04	Puede que el límite de tiempo establecido no sea suficiente para desarrollar el proyecto
Experiencia y capacidad	R06	El grupo de desarrollo no tiene experiencia en el desarrollo de software
Definición del cliente	R02	El cliente puede proponer grandes cambios para el proyecto.
Duración y tamaño	R15	La integración de todo el sistema puede llevar más tiempo del esperado
	R14	El aprendizaje de las nuevas tecnologías puede llevar más tiempo



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

		del esperado.
Duración y tamaño	R13	Las partes del sistema que no se especificaron consumen más tiempo de lo esperado
Compromiso del cliente	R01	El cliente no estará presente durante todo el proceso de desarrollo

Resumen de Riesgos

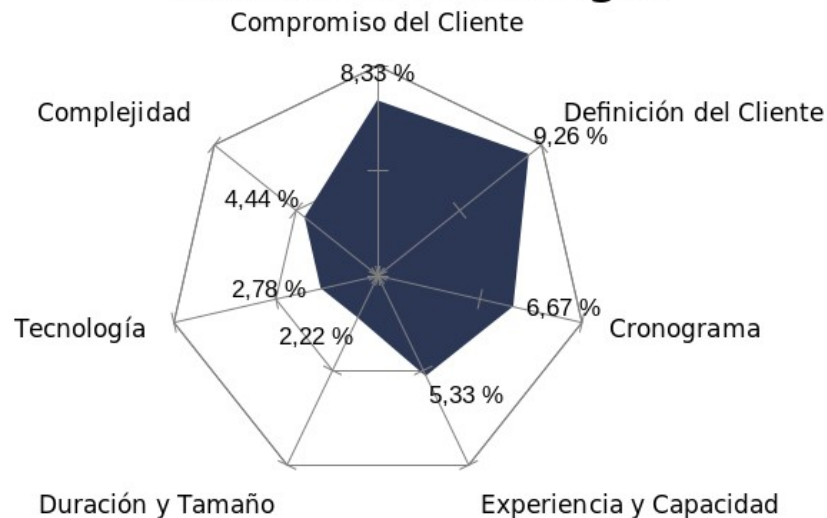


Figura 4: Distribución de los riesgos identificados en las categorías (II)

En esta segunda evaluación se consideraron 8 riesgos, se logró **mitigar el riesgo R09, R05 y R07**, el orden de prioridad de los riesgos cambio y se decide sumar a la gestión de riesgos a R4 y R2. También se identificó un nuevo riesgo con el id R15.

En la Figura 4 se puede ver como en esta segunda estimación el peso de los riesgos identificados se inclinaban por la categoría Definición del cliente. Esto se debe a que en este momento de la iteración nos encontrábamos conociendo de una manera más detallada las características del sistema, y nos encontrábamos diseñando e implementando dichas características, y se tenía la incertidumbre acerca de si los clientes pudieran pedir algo muy diferente a lo que se estaba modelando e implementado.



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

A continuación, en la Tabla 7 , se detallará que acciones se planificaron frente a estos riesgos y la actualización de los estados de los riesgos.

Tabla 7: Gestión de riesgos II

Riesgo	Acción
R4	<p><i>Estrategia de reducción:</i> realizar los planes de iteración y verificar que se cumplan con las fechas de entrega. Hacer seguimientos de los avances de cada miembro del equipo y en caso de no haberlos discutir al respecto.</p> <p><i>Estrategia de contingencia:</i> solicitar una reunión con el cliente para explicarle las causas de la no entrega del software (u artefactos) y mediar la posibilidad de una prórroga para terminar el desarrollo del sistema.</p>
R2	<p><i>Estrategia de reducción:</i> conversar con el cliente acerca de la factibilidad del cambio propuesto por su parte. Y analizar posibles alternativas que puedan satisfacer la necesidad del cliente.</p>

Tercera estimación y evaluación de riesgos — Fase 3, iteración 3

En esta nueva estimación se detectaron 2 nuevos riesgos que se sumaron a los 7 de la estimación anterior, uno a la categoría Complejidad y el otro a la categoría Definición del cliente. También cambió el orden de prioridad de riesgos como se puede ver en la Tabla 8. Se puede notar como el riesgo R06, que antes se encontraba entre los primeros, ahora se encuentra muy por debajo, lo que demuestra que el equipo de desarrollo fue adquiriendo más confianza durante el proceso de desarrollo de software lo que hizo que la prioridad del riesgo disminuyera considerablemente.

Tabla 8: Tercera estimación de riesgos

Categoría	ID Riesgo	Riesgo
Complejidad	R16	El sistema de validación de permisos de usuario puede ser complejo de implementar o incluso puede no ser posible implementarlo.
Definición del cliente	R02	El cliente puede proponer grandes cambios para el proyecto.
Cronograma	R04	Puede que el límite de tiempo establecido no sea suficiente para



		desarrollar el proyecto
Duración y tamaño	R13	Las partes del sistema que no se especificaron consumen más tiempo de lo esperado
Compromiso del cliente	R01	El cliente no estará presente durante todo el proceso de desarrollo
Duración y tamaño	R15	La integración de todo el sistema puede llevar más tiempo del esperado
Experiencia y capacidad	R06	El grupo de desarrollo no tiene experiencia en el desarrollo de software
Definición del cliente	R17	Los requerimientos definidos hasta el momento por el cliente pueden cambiar
Duración y tamaño	R14	El aprendizaje de las nuevas tecnologías puede llevar más tiempo del esperado.

Resumen de Riesgos

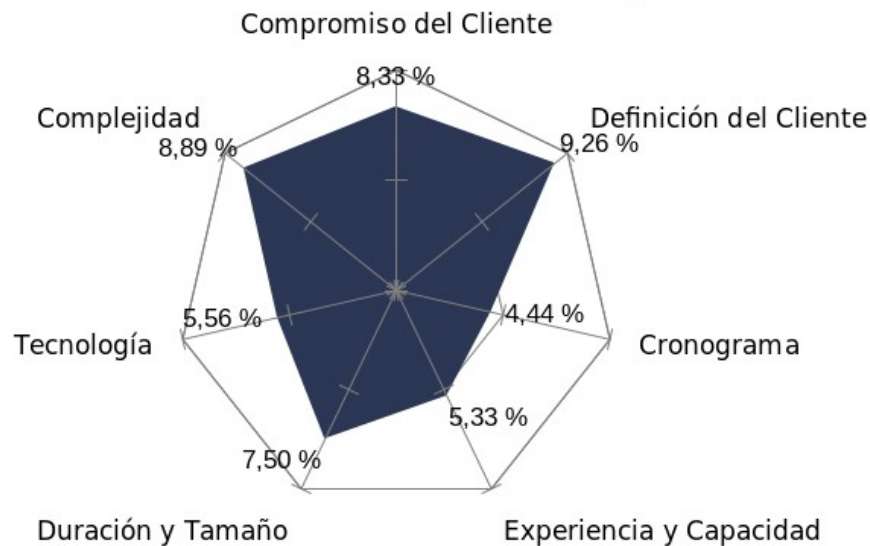


Figura 5: Distribución de los riesgos identificados en las categorías (III)



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

Se consideraron 9 riesgos, de los cuales algunos estaban siendo gestionados desde estimaciones anteriores y se sumaron a esa gestión y seguimiento los nuevos riesgos: R16 y R1. En la Tabla 9 se puede ver las estrategias adoptadas y su estado. Ningún riesgo fue mitigado.

En la Figura 5 se puede apreciar que los riesgos en esta nueva estimación se mantenían relacionados con la Definición del cliente, pero también con el Compromiso del cliente y la Complejidad. La preocupación por los riesgos relacionados con el Compromiso del cliente se debían a que nos encontrábamos hace 2 iteraciones trabajando sin mantener una comunicación tan directa y continua con los clientes, como la que manteníamos en las anteriores iteraciones, cuando cursábamos la materia en la que desarrolla este trabajo. Por otro lado en la categoría Complejidad apareció un nuevo riesgo crítico y no plantear este riesgo a los clientes y proponer alternativas podría atrasar el desarrollo de la aplicación móvil.

Tabla 9: Gestión de riesgos III

Riesgo	Acción
R16	<i>Estrategia de eliminación:</i> se definió programar una reunión con los clientes para explicarles las dificultades encontradas y presentarles posibles alternativas para abordar el problema.
R01	<i>Estrategia de reducción:</i> se definió programar reuniones con los clientes para mostrarles los avances al terminar una serie de iteraciones. También se le harán consultas vía correo electrónico de ser necesarias de manera inmediata.

Cuarta estimación y evaluación de riesgos — Fase 3, iteración 3

En esta cuarta estimación y evaluación no aparecieron nuevos riesgos, y se pasó de tener en consideración 9 riesgos de la estimación anterior a 3. Varios riesgos fueron mitigados y también bajo la probabilidad de muchos de ellos y por eso es que en esta nueva evaluación la cantidad de riesgos bajo drásticamente. Los riesgos restantes y su orden de prioridad se puede ver en la Tabla 10.

Tabla 10: Cuarta estimación de riesgos

Categoría	ID Riesgo	Riesgo
Cronograma	R04	Puede que el límite de tiempo



		establecido no sea suficiente para desarrollar el proyecto
Compromiso del cliente	R01	El cliente no estará presente durante todo el proceso de desarrollo
Experiencia y capacidad	R06	El grupo de desarrollo no tiene experiencia en el desarrollo de software

Resumen de Riesgos

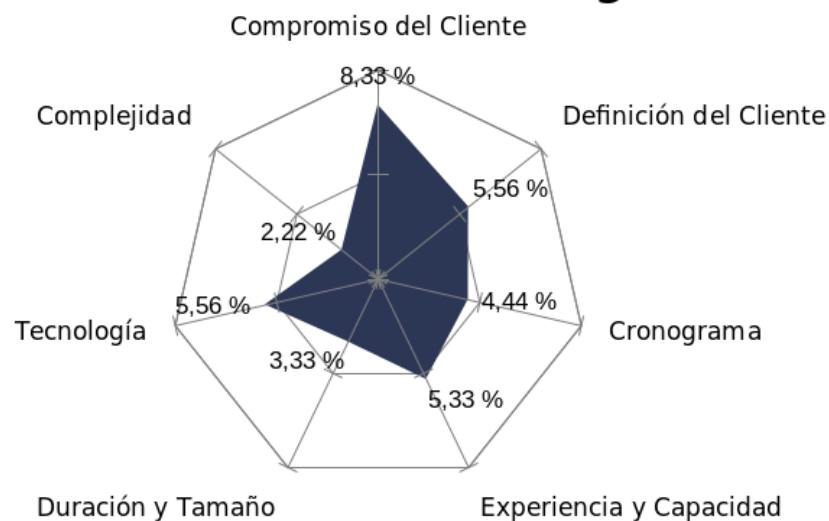


Figura 6: Distribución de los riesgos identificados en las categorías (IV)

De los 3 riesgos, dos ya se encuentran siendo gestionados y con un seguimiento activo. De los riesgos gestionados **se mitigaron los siguientes:** R02, R16. No se agregó ningún nuevo riesgo para gestionar. En la Figura 6, se puede ver una notable disminución en los riesgos de la categoría Complejidad, esto se debe a que se logró concordar con los clientes una solución para el R16 y por ende ese riesgo tiene una probabilidad de ocurrencia 0. Por otro lado, también se puede ver un leve descenso en la categoría Definición del Cliente, y esto se debe a la mitigación del riesgo R02.

Quinta estimación y evaluación de riesgos — Fase 3, iteración 8



Grupo Paire
Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

En esta quinta estimación no se identificó ningún nuevo riesgo, y los riesgos a considerar se redujo a 2 siendo R04, el primero, y R01 el segundo. Ambas riesgos ya se encontraban en seguimiento, y se estaban aplicando las estrategias de reducción.

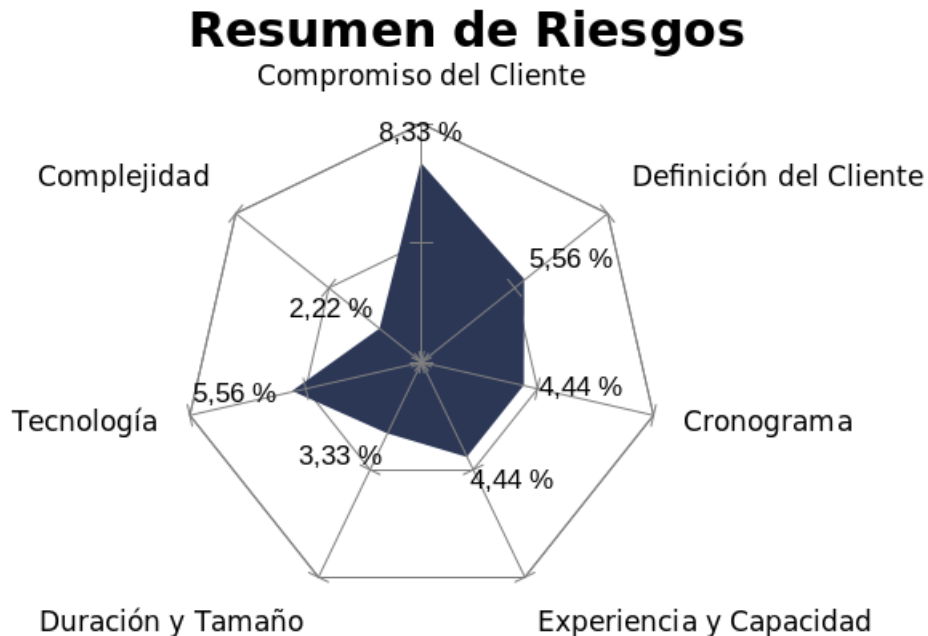


Figura 7: Distribución de los riesgos identificados en las categorías (V)

En la Figura 7 se puede ver una pequeña reducción en la categoría Experiencia y Capacidad, debido a que en esta nueva evaluación el riesgo R06, que si bien se mantiene identificado no llegó a ser considerado nuevamente porque el equipo obtuvo una gran cantidad de experiencia del proceso de desarrollo luego de realizar varias iteraciones.

Sexta estimación y evaluación de riesgos — Fase 3, iteración 11

En esta última estimación no se identificaron nuevos riesgos, y el único riesgo a considerar es el riesgo R04, el R06 no fue analizado porque debido al contexto de pandemia se extendieron las regularidades de las materias y la materia en la cual debemos hacer la entrega de este proyecto recién vence el 01/03/2021 y considerando los avances obtenidos era altamente probable que el proyecto este terminado antes de esa fecha. Esta extensión nos permitió además de dedicarnos al proyecto poder seguir avanzando académicamente, ya que contábamos con tiempo suficiente para terminar el



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

desarrollo y además al estar en confinamiento, le permitió al grupo de trabajo realizar las tareas de manera muy ágil debido a que se mantenía en constante contacto para responder rápidamente a cualquier acontecimiento y avanzar con el proyecto. Es por esta razón que en la Figura 8 la categoría cronograma se redujo totalmente.

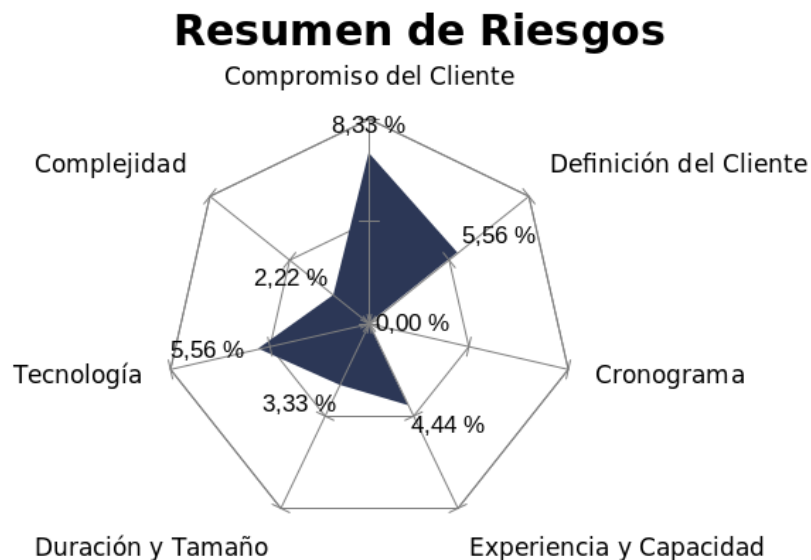


Figura 8: Distribución de los riesgos identificados en las categorías (VI)

La gestión de riesgos realizada fue de utilidad para mantener controlados los riesgos que consideramos que podrían causar un mayor impacto negativo en la planificación del proyecto. También tenemos que reconocer que la realización de las estimaciones y evaluaciones de riesgos no fueron realizadas de acuerdo a un intervalo predefinido, sino que se fueron haciendo cuando identificábamos nuevos riesgos o cuando algún riesgo era mitigado o eliminado. Hubiese sido más correcto hacer estimaciones en un tiempo predefinido. Como mencioné anteriormente, en otra actividad de aseguramiento de la calidad, la cantidad de estimaciones también depende de las características que tiene el proyecto, su tamaño y complejidad.

Validación y verificación del software

Para asegurar la validación y verificación del producto de software se redactó un plan de pruebas donde, entre otras cosas, se especificaron los



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

aspectos del sistema que se iban a probar y se definieron los distintos casos de prueba.

Los casos de prueba definidos se ejecutaron formalmente por primera vez en Noviembre del 2017 (durante la primera iteración de la fase de elaboración), y desde entonces, no se volvieron a ejecutar (formalmente) hasta Julio del 2020 (durante la décima iteración de la fase de construcción, donde también se redefinieron y agregaron otros casos de prueba para la aplicación móvil). Realmente, durante la construcción del sistema se llevaron a cabo pruebas todo el tiempo: surgieron errores a medida que se iban codificando casos de uso y dichos errores fueron solucionados. El problema es que todas esas pruebas se llevaron a cabo informalmente, es decir, no se documentaron. Debido a esto, no se registró información sobre qué tipos de errores se encontraron y cuánto tiempo y esfuerzo tomó solucionarlos.

Todo esto se debe a un error por parte del equipo de desarrollo en la gestión del proyecto. Ejecutar formalmente casos de prueba a medida que se va codificando el producto de software es importante, ya que además de servir de registro, su ejecución formal puede facilitar la tarea de encontrar el origen de los errores existentes y, en consecuencia, ahorrar tiempo en la solución de dichos errores.

Cabe destacar también que, debido a que los casos de prueba se volvieron a ejecutar formalmente en la última iteración de la fase de construcción, el resultado de tal ejecución fue “positivo” (es decir, no se encontraron errores porque ya se habían solucionado mediante las pruebas informales. En otras palabras, los casos de prueba no cumplieron su función: la de encontrar errores —Sin embargo, de todas maneras su ejecución nos permitió verificar que el sistema funcionaba correctamente).

Herramientas y tecnologías

En esta sección del documento se hará mención de las tecnologías que se utilizaron durante y para el desarrollo del proyecto.

Gestión del proyecto

- **GitHub** (github.com): GitHub es una *forja* (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

de programas de ordenador. En el proyecto, GitHub se utilizó para el control de versiones. Se creó [un repositorio para la aplicación web y la documentación del proyecto](#) y otro para la aplicación móvil.

- **Taiga** (taiga.io): Taiga es una herramienta para la gestión de proyectos, dirigida a equipos ágiles y multifuncionales. Ofrece un amplio conjunto de funcionalidades y consta de una interfaz de usuario intuitiva.

Cabe destacar que esta herramienta sólo se utilizó durante los primeros meses del proceso de desarrollo. Se dejó de utilizar ya que, al abandonar el proyecto uno de los miembros del equipo, este pasó a estar compuesto únicamente por dos integrantes, por lo que la herramienta dejó de ser de utilidad.

Clases, lenguajes y frameworks

- **CSS y HTML:** CSS es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado (como HTML). HTML es un lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, vídeos, juegos, entre otros. Gran parte de las aplicaciones web y móvil están codificadas en estos lenguajes.

- **Ionic** (ionicframework.com): El framework Ionic es un kit de herramientas móviles de código abierto para el diseño de *experiencias* de aplicaciones web y nativas multiplataforma y de alta calidad. Ionic se empleó para el desarrollo de la aplicación móvil del sistema Colibrí.

- **TypeScript** (typescriptlang.org): TypeScript es un lenguaje de programación libre y de código abierto desarrollado mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases. TypeScript puede ser usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor (Node.js y Deno). El lenguaje de programación TypeScript se utilizó para programar diversas funcionalidades de la aplicación móvil.

- **JavaScript:** JavaScript es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.



Grupo Paire

Ariel Machini y Cinthia Lima

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio es también significativo.

- **PHP** (php.net): La aplicación web del sistema Colibrí está escrita principalmente en PHP. PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web. El código PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un daemon o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado (que puede ser cualquier tipo de datos, como el HTML generado o datos de imágenes binarias) formaría la totalidad o parte de una respuesta HTTP.

- **PHPMailer** (github.com/PHPMailer/PHPMailer): PHPMailer es una clase para la creación y transferencia de correo electrónico con soporte para SMTP. La clase PHPMailer se utiliza en la aplicación web para enviar notificaciones sobre nuevas respuestas a formularios por correo electrónico.

- **TCPDF** (tcpdf.org): TCPDF es una clase PHP libre y de código abierto que permite generar documentos PDF. Esta clase se utiliza en la aplicación web para generar documentos PDF en algunos módulos de la aplicación.

- **UARGFlow**: UARGFlow es un framework desarrollado por alumnos y profesores de la UARG para la creación de aplicaciones responsivas. Uno de los objetivos principales de UARGFlow es estandarizar el aspecto y algunas de las funcionalidades de las aplicaciones de la universidad. La aplicación web del sistema Colibrí fue desarrollada con base en este framework.

Herramientas para el desarrollo

- **MySQL Workbench** (mysql.com/products/workbench): MySQL Workbench es una herramienta visual unificada para arquitectos, desarrolladores y administradores de bases de datos. Cuenta con funcionalidades para el modelado de datos, desarrollo SQL y herramientas de administración integrales para la configuración de servidores de bases de datos. En el marco del proyecto, MySQL Workbench se utilizó para la depuración y ejecución de scripts SQL.



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

- **NetBeans** (netbeans.org): Este IDE (Entorno de Desarrollo Integrado) se utilizó para el desarrollo de la aplicación web, la cual está escrita principalmente en PHP. NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo.
- **Visual Studio Code** (code.visualstudio.com): Visual Studio Code es un editor de código fuente liviano pero potente. Brinda soporte nativo para JavaScript, TypeScript y Node.js y, mediante la instalación de extensiones, permite agregar soporte para otros lenguajes como C, Java, Python, PHP o Go y para tiempos de ejecución como .NET o Unity. Esta aplicación se utilizó para el desarrollo de la aplicación móvil.

Modelado

- **DB-Main** (dataengineers.eu/en/db-main): Rever DB-Main es una plataforma de arquitectura y modelado de datos que permite, entre otras cosas, realizar diagramas entidad-relación. Esta herramienta se utilizó para diseñar diagramas entidad-relación (presentes en el modelo de datos) para la construcción de la base de datos del sistema Colibrí.
- **UMLet** (umlet.com): UMLet es una herramienta libre y de código abierto para la creación de diagramas UML, así como su posterior exportación en diversos formatos (EPS, PDF, JPG, SVG, entre otros). UMLet se utilizó para crear muchos de los gráficos presentes en la documentación del proyecto.

Otros

- **Apache** (httpd.apache.org): El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616. Esta herramienta se utilizó para alojar localmente el sistema Colibrí.
- **LibreOffice** (es.libreoffice.org): LibreOffice es un paquete de software de oficina libre y de código abierto desarrollado por The Document Foundation. Se creó en 2010 como bifurcación de OpenOffice.org, otro antiguo proyecto de código abierto. El paquete de ofimática LibreOffice se utilizó a lo largo de todo el proceso de desarrollo para escribir la documentación (tanto documentos de texto como hojas de cálculo y presentaciones) del proyecto.



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

- **MySQL** ([mysql.com](https://www.mysql.com)): MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation. Está considerada como la base de datos de código abierto más popular del mundo y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web. El sistema de gestión de bases de datos MySQL se utilizó para la puesta en funcionamiento de las bases de datos requeridas para el funcionamiento del sistema Colibrí.

- **VirtualBox** ([virtualbox.org](https://www.virtualbox.org)): Oracle VM VirtualBox (conocido generalmente como VirtualBox) es un software de virtualización para arquitecturas x86/amd64. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como *sistemas invitados*, dentro de otro sistema operativo *anfitrión*, cada uno con su propio ambiente virtual. VirtualBox se utilizó para crear máquinas virtuales sobre las cuales se desplegaron entornos de desarrollo adecuados para la aplicación móvil/web.

- Máquina virtual utilizada para el desarrollo de la aplicación web:
 - **SO:** Xubuntu 18.04.
 - **Almacenamiento:** 35 GB.
 - **Cantidad de procesadores:** 4 (virtuales).
 - **Memoria RAM:** 8000 MB.
 - **Tipo de adaptador de red:** Puente.
- Máquina virtual utilizada para el desarrollo de la aplicación móvil:
 - **SO:** Xubuntu 18.04.
 - **Almacenamiento:** 10 GB.
 - **Cantidad de procesadores:** 2 (virtuales).
 - **Memoria RAM:** 2000 MB.
 - **Tipo de adaptador de red:** Puente.

Desarrollo del proyecto

En esta sección del documento se detallarán las actividades, acontecimientos importante o problemas que existieron en las cuatro fases del proceso de desarrollo. Cada fase estuvo compuesta por un conjunto de iteraciones que también serán mencionadas y explicadas. Luego de explicar cada una de las fases se hará un análisis global de todas las iteraciones en términos de duración en días de cada una, cantidad de actividades cumplidas y cantidad de actividades cumplidas así como también un



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

análisis del porcentaje de cumplimiento de las actividades en las iteraciones.

Primera fase: Fase de inicio

En la primera fase del proceso de desarrollo el equipo contaba con muy poca experiencia y, debido a esto, sólo se planificó una única iteración para toda esa fase. En dicha iteración se planificó la redacción de diversos documentos de gran importancia para el proceso de desarrollo. Nuevamente, debido a la inexperiencia del equipo (además de varias dudas que surgieron en torno a la documentación), ninguno de los documentos pudo ser completado en su totalidad para la fecha de cierre de la iteración planificada.

Cabe destacar también que en esta etapa el equipo de desarrollo no estaba compuesto por dos personas, si no por tres.

INFORMACIÓN SOBRE LA FASE DE INICIO	
Duración (días)	18
Cantidad de iteraciones	1
Cantidad de actividades	5
Cantidad de actividades terminadas a tiempo	0 de 5 (0%)

Segunda fase: Fase de elaboración

Antes de comenzar a detallar la fase de elaboración hay que realizar una aclaración importante: En este punto, uno de los integrantes del equipo de desarrollo se vio obligado a abandonar el proyecto a raíz de complicaciones personales, por lo tanto, a partir de esta iteración, el equipo pasó a estar conformado por dos personas.

En esta fase del proceso de desarrollo el equipo todavía contaba con poca experiencia por lo que, similar a lo que aconteció en la fase de inicio, durante el transcurso de esta fase sólo se planificó una iteración (¡una *gran* iteración!). Sin embargo, a diferencia de lo que sucedió en la iteración anterior, en esta iteración se pudo completar la mayoría de las tareas planificadas. Cabe destacar que en esta fase se comenzó con la implementación de algunos de los casos de uso de la aplicación web del sistema Colibrí, sin embargo, la mayoría de las actividades que se



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

planificaron durante esta fase estuvieron enfocadas en el diseño o en la redacción de documentos.

Un detalle menor: Desde esta fase en adelante, en los planes de iteración se comenzó a incluir más información (la fecha de cierre de la iteración y los miembros responsables de las tareas) para facilitar el seguimiento de las tareas.

INFORMACIÓN SOBRE LA FASE DE ELABORACIÓN	
Duración (días)	54
Cantidad de iteraciones	1
Cantidad de actividades	23
Cantidad de actividades terminadas a tiempo	18 de 23 (78%)

Tercera fase: Fase de construcción

La fase de construcción fue la más extensa de todas las fases del proceso de desarrollo y su enfoque principal fue en torno a la implementación de las aplicaciones del sistema Colibrí.

La primera iteración de esta fase se planificó cerca del final de la cursada (en el año 2017), abarcó una gran cantidad de tareas (muchas de ellas de suma importancia, como la implementación de casos de uso esenciales del sistema) y su éxito fue crucial para el desarrollo del proyecto: Gracias a ello, los miembros del equipo regularizaron la materia (Laboratorio de Desarrollo de Software) y pudieron continuar con el proceso de desarrollo del sistema Colibrí. Tras cerrar esta iteración, la cursada finalizó y el tiempo pasó: Por desgracia, los miembros del equipo no pudieron seguir el ritmo que iban llevando durante la cursada debido a la pesada carga por la que pasaron ese cuatrimestre así como también porque tuvieron que cumplir con otras responsabilidades. Debido a esto, la siguiente iteración se planificó (después de una reunión entre los miembros del equipo) poco más de un año después (en Enero de 2019) de haber cerrado la última iteración. En esa nueva iteración se retomó la redacción de documentos que habían quedado pendientes, se inició el proceso de actualización de UARGFlow¹ (a UARGFlow BS) y la implementación de las aplicaciones del

¹ Se tomó la decisión de actualizar la versión de UARGFlow en 2018, tras una conversación con los clientes (los docentes de la materia). Dicha actualización implicaría la reimplementación de los casos de uso ya implementados (de la



sistema (móvil y web) se distribuyó entre los miembros del equipo de desarrollo.

La tercer iteración de la fase de construcción se planificó unos meses después de la anterior, tras haber tenido una reunión con los profesores. Se considera que esta iteración fue muy importante ya que, a partir de una recomendación de uno de los profesores, se comenzarían a organizar iteraciones más cortas y con menos actividades. Gracias a que se siguió dicha recomendación, esta fue la primera iteración en la que se pudieron terminar todas las tareas planificadas a tiempo. Definir objetivos realizables en períodos de tiempo razonables permitió lograr iteraciones más productivas sin decrementar el ritmo de trabajo, cosa que impactó positivamente en la motivación de los miembros del equipo de desarrollo. Como se escribió anteriormente, debido al éxito de esta técnica, las siguientes iteraciones se planificaron de la misma manera.

Tras cerrar la quinta iteración, los miembros del equipo de desarrollo tuvieron que dejar de lado nuevamente el desarrollo del proyecto ya que tenían que cumplir con otras responsabilidades académicas (entre ellas, preparar o rendir algunos exámenes finales). Transcurridos nueve meses (en Marzo de 2020) y tras una reunión con los profesores, se planificó la sexta iteración de la fase de construcción. Cabe mencionar que en esta iteración se realizaron algunas modificaciones en el diseño del sistema debido a ligeros cambios en los requerimientos propuestos por los clientes. En el transcurso de esta iteración también se tomó la decisión de crear un repositorio aparte para la aplicación móvil (separado del repositorio en el que se encuentra la documentación del proyecto y la aplicación web).

Durante la séptima iteración se descubrió un problema con la manera en la que se recuperaban los campos de los formularios del sistema en la aplicación móvil, por lo que se tuvo que idear y codificar una forma alternativa para recuperarlos. En la siguiente iteración se probó la alternativa y resultó funcionar, por lo que el problema quedó solucionado. Tras cerrar la novena iteración y, por supuesto, al haber completado todas las tareas planificadas en dicha iteración, la primera versión completa del sistema Colibrí (tanto la aplicación web como la aplicación móvil) había quedado terminada. Este hito permitió al equipo de desarrollo comenzar con la culminación de la fase de construcción del proceso de desarrollo.

La última iteración a destacar en la fase de construcción es la décima y postrera: Dicha iteración estuvo enfocada en la redefinición y ejecución de

aplicación web).



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

casos de prueba para verificar el correcto funcionamiento del sistema Colibrí antes de continuar con la siguiente fase del proceso.

INFORMACIÓN SOBRE LA FASE DE CONSTRUCCIÓN	
Duración (días)	161
Cantidad de iteraciones	10
Cantidad de actividades	72
Cantidad de actividades terminadas a tiempo	59 de 72 (82%)

Cuarta fase: Fase de transición

Durante la fase de transición se redactaron los últimos documentos requeridos para finalizar con el proceso de desarrollo del proyecto: el manual de instalación, el manual de usuario y este documento (la memoria del proyecto). Para el desarrollo de estos documentos se obtuvo retroalimentación por parte de los profesores de la materia. En total, se planificaron tres iteraciones para terminar dichos documentos al cien por ciento.

INFORMACIÓN SOBRE LA FASE DE TRANSICIÓN	
Duración (días)	29
Cantidad de iteraciones	3
Cantidad de actividades	10
Cantidad de actividades terminadas a tiempo	10 de 10 (100%)

Sumando la duración de todas las fases, en total, el proceso de desarrollo del proyecto Colibrí tomó **262 días de trabajo**, lo que equivale aproximadamente a 8 meses y medio.

A continuación en la Figura 9 se muestra un resumen de la cantidad de días empleados en cada iteración, y también se puede distinguir las iteraciones realizadas en casa fase. La iteración con mayor cantidad de días fue la número 1 de la fase de Elaboración con 54 días debido a la poca experiencia del equipo la duración y cantidad de tareas programadas era muy elevada. Algo similar sucedió en la iteración 2 de la fase de



Construcción ya que la iteración duro 48 días. Luego de esa iteración la duración de los días de comenzó a normalizar, debido a los consejos de los profesores como se menciona anteriormente. En términos generales la duración de las iteraciones fue evolucionando de acuerdo al estado del proyecto y las necesidades de los clientes, al principio la duración era un poco más arbitraria y las iteraciones no eran tan eficientes, pero la duración se fue ajustado una vez que se fue conociendo el ritmo de trabajo del equipo (y por recomendación de los profesores), de manera que el resultado de cada iteración logró ser más eficiente y productivo.

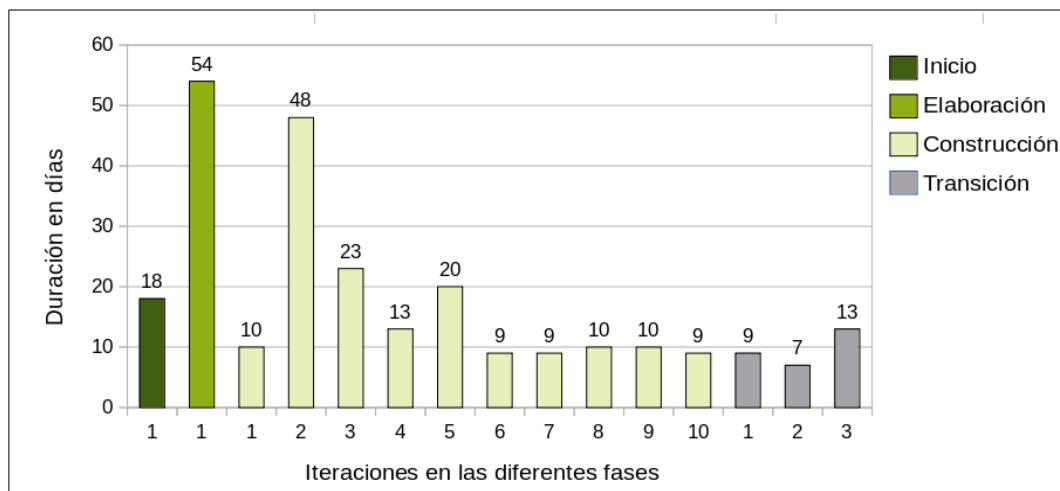


Figura 9: Duración de todas las iteraciones agrupadas por fase

En la Figura 10 se pueden ver la cantidad de tareas definidas en cada iteración y que cantidad de ellas fueron cumplidas. Se puede ver que la iteración con mayor cantidad de tareas fue la número 1 de la fase de Elaboración, que coincide también con la iteración de mayor duración como se puede notar en la Figura 9. El numero elevado de tareas tiene relación con las tareas pendientes de la iteración anterior. También se puede notar que en las primeras 4 iteraciones que las tareas planificadas nunca lograron ser completas en su totalidad. Recién a partir de la segunda iteración de la fase de Elaboración se distingue un cambio drástico en la cantidad de tareas cumplidas y también una reducción en la cantidad de tareas planificadas.

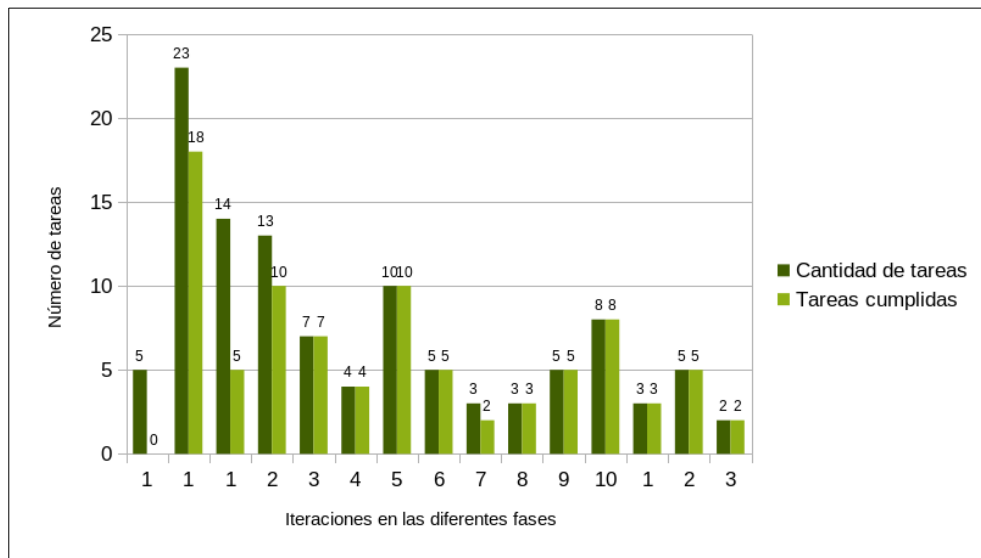


Figura 10: Tareas planificadas vs Tareas cumplidas

En términos de efectividad a la hora de cumplir con el plan de iteración, teniendo en cuenta las 15 iteraciones pertenecientes a las diferentes fases, únicamente existieron 5 iteraciones en las cuales no se puede cumplir con lo planificado en el plan de iteración. Se puede ver en la Figura 11 cuales son esas iteraciones, y representan el 33% del total de la iteraciones. Las iteraciones en las que existió un cumplimiento de las actividades menor al 50% fueron dos: la primera de la fase de Inicio y la primera de la fase de Construcción.

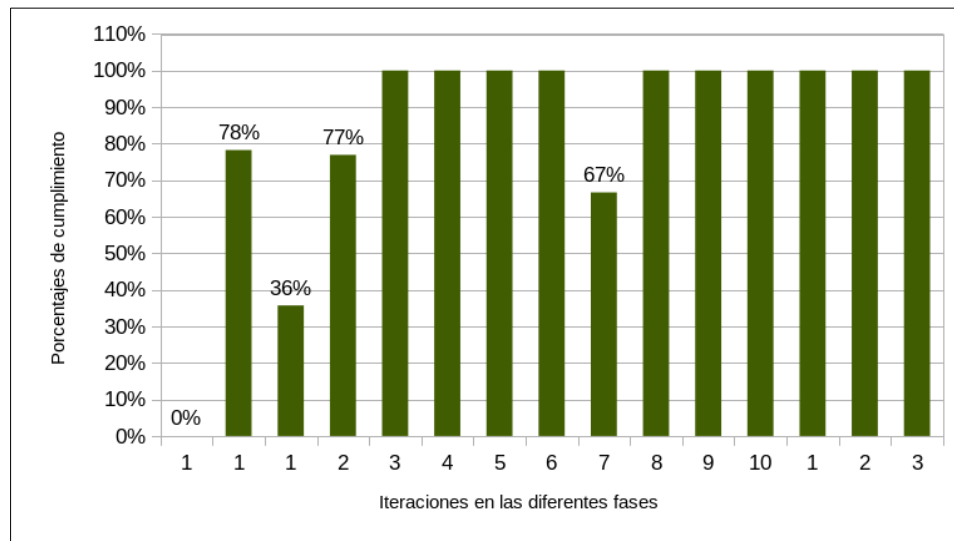


Figura 11: Nivel de cumplimiento de las actividades planificadas en el Plan de iteración

En la primera de la fase de Inicio donde no se logró finalizar con ninguna de las tareas propuestas, que como se mencionó en la sección correspondiente a la fase, no significa que no se trabajó en la iteración sino que los entregables estaban muy avanzados, pero no estaban del todo completos. En la primera iteración de la fase de Elaboración se pueda notar un nivel de cumplimiento del 36%, en esta fase se dejaron muchos pendientes porque para la regularización de la materia nos solicitaron ciertos entregables e implementaciones, entonces el equipo se avocó exclusivamente las actividades relacionadas con lo requerido, dejando muchos pendientes. El no cumplimiento de los actividades en la primera iteración del desarrollo, acarreo complicaciones a las siguientes iteraciones donde además de tener que planificar nuevas actividades también se tenían que terminar la de fases anteriores, esto se puede notar en las primeras 4 iteraciones (hablando en términos generales sin distinguir fases) que no logran finalizar esa cadena de artefactos incompletos, si no es recién en la tercera iteración de la fase de Construcción. Por otro lado en la iteración 7 de la fase de construcción en la cual sólo se logró completar el 67% de las actividades planificadas se debe a que no se logró terminar por completo la implementación de un caso de uso en la aplicación móvil por falta de tiempo, y porque se tuvo que investigar y experimentar para poder elegir un método de validación de campos que se adapte al problema.

Conclusión

Como se explicó a lo largo de la memoria, el proceso de desarrollo del software implicó varias tareas en las diferentes disciplinas de la ingeniería del software. Redactar esta memoria nos permitió dejar en evidencia todo el esfuerzo y tiempo dedicado para el desarrollo de este producto de software, las actividades que tuvimos que llevar a cabo para lograrlo y las dificultades que tuvimos y como los afrontamos.

No cabe duda que el desarrollo de este proyecto y todo lo que implicó nos dio nuestro primer acercamiento con el desarrollo de software de calidad. Aplicamos todo lo aprendido en diferentes materias de la carrera, lo cual hizo que sea en parte satisfactorio poder emplear de manera integral todo ese conocimiento adquirido durante años de cursada en una práctica real. Esta primera experiencia nos dejó varias enseñanzas como así también nos permitió conocer nuestras capacidades individuales y como equipo. En relación a las enseñanzas aprendidas pudimos comprender con mayor claridad las tareas relacionadas con el desarrollo de software desde el análisis del problema a resolver hasta una implementación efectiva que lo resuelva, también pudimos apreciar que la gestión del proyecto es una tarea muy importante y al mismo tiempo demandante pero que es necesaria para poder organizar las tareas, actividad fundamental en el desarrollo de cualquier tipo de proyecto, y más en el desarrollo de software, ya que por lo general es una actividad iterativa que constantemente requiere supervisión para conocer el estado del proyecto en cada momento y no perder el control del mismo. Sin la existencia de una gestión el desarrollo de este proyecto no hubiese sido el mismo, los tiempos no hubiesen sido los mismos y las tareas se hubiesen realizado en diferente orden. La gestión de los riesgos también debe destacarse ya que sin la detección temprana de los riesgos, su seguimiento y definición de estrategias de mitigación para estos, el tiempo necesario para el desarrollo hubiese sido más extenso. Esto se puede apreciar en la sección de gestión de riesgos, donde especificamos que riesgos gestionamos y cuales estrategias adoptamos.

Las estimaciones y planificaciones previstas nos fueron de utilidad para avanzar con la confianza de que íbamos por el camino correcto y que los tiempos estipulados nos eran suficientes. Por suerte no tuvimos contratiempos con la entrega, como explicamos en la sección de estimación y planificación.

En términos de comunicación podemos detectar dos grandes relaciones presente durante todo el desarrollo. La primera es la relación del equipo de



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

trabajo, la cual fue muy amena y coincidimos en que si bien, tuvimos algunas largas conversaciones para la toma de las diferentes decisiones, ya sea de diseño, implementación o planificación, entre otras, pudimos llegar a acuerdos que complacieran al grupo y sobre todo que dichas decisiones impactaran positivamente en el producto final. Por lo general las decisiones tenían que ver con preferencias particulares, que obviamente cada persona tiene y no siempre coinciden, lo importante de estas diferencias es poder afrontarlas y solucionarlas en equipo para así poder llegar a la mejor opción para el cliente. Otra de las relaciones fue la que se mantuvo con los clientes, que coincide también con los profesores de la cátedra en la que desarrollamos este proyecto. Fue nuestro primer acercamiento con un cliente real, pudimos al fin poder hacer la toma de requerimientos reales y todo lo que implica, luego de haber practicado en tanta materias dicha actividad pero con un escenario simulado. Pudimos apreciar la importancia de mantener una buena comunicación con el cliente, ya que nos ayudo mucho para conocer la evolución de la satisfacción del mismo, así como también para avanzar de manera segura en las tareas del desarrollo. Gracias a la constante comunicación mantenida, ellos pudieron expresar sus impresiones acerca del trabajo que se iba realizando y sugerir algunos cambios de requerimientos o expresar su satisfacción con alguna funcionalidad implementada.

El desarrollo de este proyecto también nos permitió aprender nuevos lenguajes y tecnologías que no conocíamos o no utilizábamos con frecuencia. También gracias a ese aprendizaje, logramos descubrir que nuestra curva de aprendizaje, debido a toda la preparación en el área de programación de la carrera, fue relativamente baja lo cual resulta un dato muy importante para nosotros como desarrolladores. El aprendizaje de estas tecnologías nos capacito para posibles desarrollos futuros en los que se usen dichas tecnologías.

Tenemos que reconocer que tuvimos una falta de atención en la pruebas realizadas, debido a que en un principio se realizaron de acuerdo a lo planificado, pero durante las iteraciones siguientes no. Cuando decimos que no, nos referimos a que no fueron documentadas, ya que obviamente si existieron pruebas para comprobar el correcto funcionamiento de lo implementado y se solucionaron errores encontrados. No realizar una debida documentación no nos permitió tener asentado que errores ocurrieron, que esfuerzos fueron necesarios para resolverlos y cuanto tiempo nos tomo resolverlo. La documentación de los errores, nos podría servir para proyectos futuros y también para la etapa de mantenimiento.



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de desarrollo de software

Consideraremos esta falta de verificación formal del correcto funcionamiento del sistema para futuros desarrollos.

También existieron problemas relacionados con la planificación de la tareas en la primera parte del desarrollo del proyecto, los tiempos no eran los adecuados y/o definíamos muchas actividades, pero gracias a las recomendaciones de los profesores pudimos definir un ritmo y carga en las iteraciones que mantuvimos durante todo el desarrollo y fue de beneficio ya que nuestras iteraciones eran más productivas. La obtención de experiencia a medida que se iba a ejecutando cada iteración nos permitía llevar las tareas de gestión del proyecto con mayor naturalidad. En relación a esto queremos también destacar y agradecer el constante acompañamiento que tuvimos por parte de los docentes que nos pudieron guiar en el desarrollo. Más allá de seguir la metodología de desarrollo PSI, que también nos fue de ayuda ya que brindaba mucha información acerca del proceso y los entregables necesarios que forman parte del producto de software, la participación, apoyo y motivación brindada por los docentes fue fundamental para poder terminar con este desarrollo.

En conclusión, el desarrollo del proyecto Colibrí nos dejó una experiencia completa del proceso de desarrollo de software, logrando gracias a esto desarrollar un producto de software para responder a un problema real. Y todo lo aprendido durante este proceso nos será útil para futuros proyectos.

