

Memoria



Proyecto Colibrí

Laboratorio de desarrollo de software

Grupo Paire

Ariel Machini

Cinthia Lima

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.



Índice

Introducción.....	4
Objetivos.....	5
Objetivo general.....	5
Metodología de desarrollo.....	5
Organización general del trabajo.....	6
Comunicación del equipo.....	6
Calidad y gestión del proyecto.....	7
Estimación y planificación del proyecto.....	7
Gestión de riesgos.....	13
Validación y verificación del software.....	19
Herramientas y tecnologías.....	20
Gestión del proyecto.....	20
Lenguajes y frameworks.....	21
Herramientas para el desarrollo.....	22
Modelado.....	23
Otros.....	23
Desarrollo del proyecto.....	24
Primera fase: Fase de inicio.....	24
Segunda fase: Fase de elaboración.....	25
Tercera fase: Fase de construcción.....	26
Cuarta fase: Fase de transición.....	27
Conclusión.....	31



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 Analista de Sistemas. La descripción del proceso se abordará desde una perspectiva mediante la cual se exponga cómo el equipo de desarrollo pudo llevar a cabo todas las tareas necesarias para la construcción del sistema. Más allá de la documentación técnica, la cual es inherente al desarrollo de este proyecto, lo que quiere mostrar esta memoria es cómo se vivió el proceso de desarrollo; qué problemas se encontraron y cómo se solucionaron y qué conocimiento o habilidades se obtuvieron gracias al desarrollo de este proyecto.

Colibrí consta, principalmente, de una aplicación web con la funcionalidad para crear y gestionar formularios altamente personalizables, así como también con la funcionalidad para que dichos formularios puedan ser completados desde la misma aplicación web o desde una aplicación móvil. Una característica a destacar sobre la aplicación web es que trabaja sobre el *framework* UARGFlow BS, el cual permite hacer una distinción entre los diferentes tipos de usuarios que utilizan el sistema. Gracias a ello, se puede definir a qué tipos de usuarios estarán dirigidos los formularios que se creen en el sistema.

El sistema Colibrí fue solicitado para su puesta en funcionamiento en la Unidad Académica de Río Gallegos de la Universidad Nacional de la Patagonia Austral ya que varios formularios de las diversas áreas de la universidad no se hallan digitalizados. Los formularios digitalizados resultan más convenientes que los impresos en papel, ya que su distribución y la recopilación de respuestas es más fácil con un formulario digitalizado.

Para llevar a cabo el desarrollo del sistema, se empleó un marco de trabajo que permite planificar, organizar y controlar todas las actividades necesarias para cumplir con el objetivo: Se utilizó la metodología de desarrollo PSI¹, la cual brindó al equipo el marco necesario para el desarrollo de las actividades. PSI incluye una colección de plantillas y ejemplos para la documentación del 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 proceso de desarrollo; como se llevó a cabo en la gestión del proyecto; las herramientas y

1 <https://www.uarg.unpa.edu.ar/psi>



tecnologías que se utilizaron; el desarrollo del proyecto (distribuido a lo largo de 4 fases) y, por último, las conclusiones de esta memoria.

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.

Objetivos específicos

- Desarrollar una aplicación web que comprenda todas las funcionalidades solicitadas por el cliente.
- Desarrollar una aplicación móvil mediante la cual se puedan completar 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 está basada en el Proceso Unificado, que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. La metodología PSI fue desarrollada como proyecto final de la carrera Licenciatura en Sistemas por algunos de los profesores de la materia.

PSI brinda una pauta general de cómo llevar a cabo el proceso de desarrollo: indica qué actividades realizar y cómo, cuándo y por qué realizarlas. Además, ofrece un conjunto detallado de plantillas, las cuales se encuentran 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 son una guía para la correcta redacción de la documentación del proyecto.

La documentación es fundamental para las etapas de mantenimiento del sistema así como también para analizar el proceso de desarrollo y, de esa manera, poder mejorar continuamente en los procesos realizados.



El proceso de desarrollo que propone PSI se divide en 4 fases: Inicio, Elaboración, Construcción y Transición. En cada fase se desarrollan un conjunto n de iteraciones, las cuales dan como resultado un incremento del producto desarrollado. Dicho incremento puede ser un artefacto o una implementación que agrega o mejora a las funcionalidades del sistema en desarrollo. A su vez, en 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 planifique 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, el equipo de desarrollo estuvo conformado por tres miembros. Pasado ese mes, por motivos de fuerza mayor el equipo redujo a sólo dos integrantes y así se mantuvo durante el resto del proceso de desarrollo.

Para completar todas las actividades del proceso, se planificaron en total 15 iteraciones, en las cuales se definieron distintas actividades que se distribuyeron entre los miembros del equipo de desarrollo. Todas estas iteraciones se documentaron en planes de iteración, los cuales van acompañados de un plan más general, el plan de proyecto.

Cada una de las iteraciones contó con una fecha de cierre, en la cual el equipo de desarrollo comprobaba sin falta qué actividades se pudieron cumplir, cuáles no y porqué. Esto quedaba documentado en el plan de la iteración junto a una conclusión sobre la iteración. Redactar planes de iteración durante todo el proceso de desarrollo fue de gran ayuda para la organización y el control de las actividades que se realizaron.

Comunicación del equipo

La buena comunicación entre los miembros del equipo de desarrollo es clave para poder organizar bien el trabajo.

Durante el desarrollo de este proyecto, los miembros del equipo convivieron juntos la mayoría del tiempo y, por ende, la comunicación entre ellos solía desarrollarse en persona y al momento en el que fuese necesaria. La buena comunicación entre los miembros del equipo permitió que discusiones en torno a los diferentes aspectos del proyecto pudieran realizarse de manera rápida y efectiva. Cabe destacar también que existieron un par de iteraciones que tuvieron que realizarse de manera virtual ya que, al momento de



Grupo Paire

Ariel Machini y Cinthia Lima

planificarlas, el los miembros del equipo de desarrollo no se encontraban conviviendo. Para planificar dichas iteraciones, la comunicación se llevó a cabo a través del servicio de mensajería instantánea WhatsApp.

Calidad y gestión del proyecto

La gestión del proyecto es una actividad esencial en el desarrollo de software que debe estar presente antes de iniciar cualquier actividad técnica y debe aplicarse durante todo el proceso de desarrollo. Una falta de gestión puede llevar al fracaso a cualquier proyecto de software, por lo que es muy importante aplicarla correctamente desde el inicio del proyecto.

Uno de los objetivos principales de la ingeniería del software es generar software de calidad, por lo tanto, debe existir una gestión de la 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.

Durante el desarrollo del proyecto realizaron actividades para el aseguramiento de la calidad, sin embargo, estas no fueron explicitadas en un plan de calidad. Es por esta razón que en una de las iteraciones finales del proceso se tomó la decisión de expresar en un plan de calidad qué actividades de aseguramiento y control se realizarían para producir un sistema de alta calidad. Reconocemos que el plan de calidad debería haberse escrito con anterioridad, pero al principio del desarrollo (durante la cursada) no fue propuesto como un entregable obligatorio. Es por esto que se redactó más adelante, cuando se contaba con más tiempo libre para redactarlo.

Las actividades que se especificaron en el plan de calidad fueron las que siguen:

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

Estimación y planificación del proyecto

La estimación y la planificación son dos actividades importantes, que se deben realizar 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.

La estimación del tamaño y esfuerzo necesario para llevar a cabo el proyecto de software es una de las primeras actividades a realizar, y para ello se debe



Grupo Paire

Ariel Machini y Cinthia Lima

estudiar el problema a resolver así como también definir su alcance, las características del entorno y del equipo de trabajo. Para recopilar toda la información para realizar la estimación se hizo un estudio de factibilidad; un modelado de negocio; 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 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. Las primeras tres iteraciones se desarrollaron en el marco de la cursada de la materia, sin tener una previa estimación. La primera estimación se llevó a cabo tras cerrar la tercera iteración del proceso de desarrollo. Para entonces, el equipo tenía una visión más clara del sistema a desarrollar, conociendo qué actividades tenían pendientes y cuáles 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.

A lo largo de todo el proceso de desarrollo se realizaron tres estimaciones, y a continuación se describirán los resultados de cada estimación y en qué condiciones se realizaron:

- **Estimación 1** (12 meses trabajando 12 horas por semana): La primera estimación se hizo el 24 de Noviembre del 2017. El estado del proyecto estaba avanzado, ya que hasta ese entonces se habían realizado tres iteraciones con muchas actividades.
- **Estimación 2** (3,5 meses trabajando 21 horas por semana): La segunda estimación se realizó el 22 de Marzo del 2020, y se hizo tras retomar el proyecto luego de trabajar en algunas iteraciones durante el año 2019. Desde la última estimación hasta esta se llevaron a cabo cuatro iteraciones (durante el año 2019), en las cuales se logró avanzar considerablemente en el desarrollo y también en los conocimientos y capacidades del equipo; Es por esto que se puede notar una gran diferencia en la duración de esta estimación en comparación con la primera. Esto se puede ver en la Figura 1, donde la línea verde oscuro representa el tiempo de esta estimación en comparación con la primera.
- **Estimación 3** (1,5 meses, trabajando 14 horas por semana): Esta estimación se hizo el 4 de Mayo del 2020. Para esa fecha ya se habían llevado a cabo cuatro iteraciones más desde la última estimación. A esa altura, el proyecto ya se encontraba en un estado bastante avanzado 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 con los tiempos estimados para una sola persona.

Observando la figura, se puede afirmar que la diferencia de tiempo cuando la cantidad de personas es mayor a uno es grande. Sin embargo, aunque parezca que sumar personal siempre permitirá obtener tiempos más bajos, esto dependerá de cómo se gestione dicho personal: Agregar más personas no siempre es la solución.

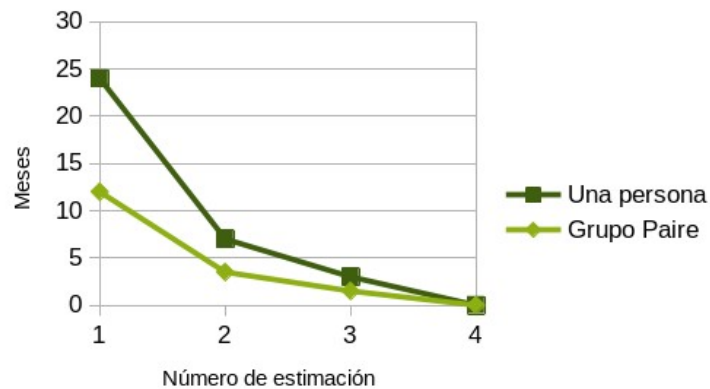


Figura 1: Estimación del proyecto.

La duración real del desarrollo en comparación con las estimaciones realizadas fue mucho menor. En la Figura 2 se puede observar que todas las estimaciones fueron mayores a la duración real del proyecto, considerando el tiempo desde esa estimación hasta el fin del desarrollo. Por otro lado, se puede notar que la diferencia entre la duración estimada y la duración real disminuía en cada nueva estimación que se realizaba. Esta disminución ocurrió ya que en cada nueva estimación realizada se contaba con información más precisa y, además, el estado del proyecto era más avanzado.

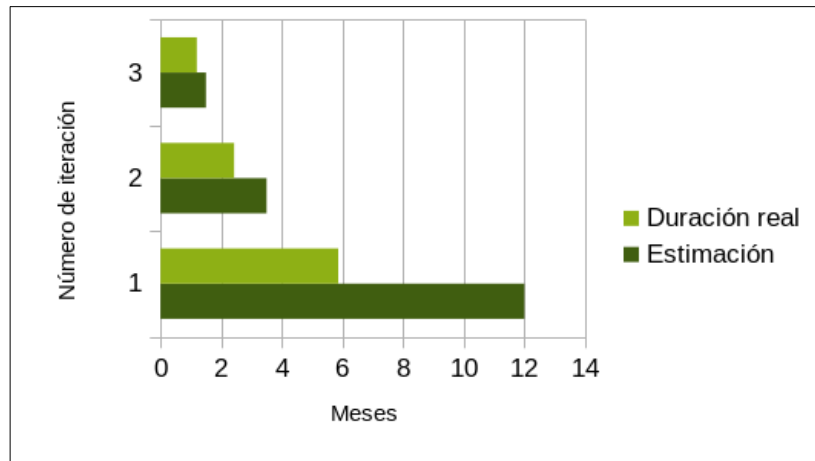


Figura 2: Comparación de la estimación con la duración real del proceso de desarrollo.

La planificación forma parte de la gestión efectiva del proyecto. Todas las actividades de ingeniería que formaron parte de iteraciones se planificaron detalladamente y su progreso se supervisó para verificar su cumplimiento. Para ello, se elaboraron planes para cada iteración, en los cuales se especificaron los recursos necesarios (humanos y materiales) para su realización, se dividieron y asignaron las actividades entre los miembros del equipo y se definieron calendarios de trabajo (tiempos). Estos planes de iteración a su vez, alimentan al plan de proyecto, en el cual se puede consultar el estado del proyecto en un momento dado. En total se planificaron 15 iteraciones, de las cuales 3 fueron realizadas en el año 2017; 4 en el 2019 y 8 en el 2020. Durante el año 2017 se trabajó en el proyecto dentro del marco de la materia Laboratorio de Desarrollo de Software, por lo que se mantenían iteraciones cargadas de actividades y, al mismo tiempo, el equipo se familiarizaba con la puesta en práctica de la metodología PSI y con el proceso de desarrollo de software en general. La cantidad de iteraciones, su duración y a qué fase pertenecen se puede ver en la Tabla 1.

Tabla 1: Iteraciones planificadas durante el año 2017.

Iteración	2017			
	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE
F1I1				
F2I1				
F3I1				
Días		18	54	10

Luego de terminar la cursada en el año 2017, el desarrollo del proyecto recién se retomó en 2019. Esto se debe a que durante el año 2018 nos encontrábamos ocupados cursando varias materias; rindiendo finales pendientes y realizando otras actividades académicas, cosa que no nos permitía trabajar en el proyecto.

En Enero del 2019 retomamos el desarrollo del proyecto: Trabajamos casi hasta mitad de año con interrupciones en Marzo y Abril. Con respecto a la mitad restante del año, nuevamente tuvimos que dejar de lado el proyecto porque (otra vez) nos encontrábamos muy ocupados con otras actividades académicas y adelantando materias. El desarrollo de las iteraciones durante el año 2019 se puede ver en la Tabla 2.

Tabla 2: Iteraciones planificadas durante el año 2019.

Iteración	2019						
	ENERO	FEB.	MARZO	ABRIL	MAYO	JUNIO	JULIO
F3I2							
F3I3							
F3I4							
F3I5							
Días		48			23	13	20

En Marzo del 2020 retomamos el proyecto, en el estado que había quedado en Julio del 2019. Durante el 2020 trabajamos desde Marzo hasta Agosto, prácticamente sin interrupciones a excepción de unos días en Mayo (en los que realizamos un seminario) y todo el mes de Junio (durante el que nos encontrábamos culminando con el cuatrimestre y trabajando en un trabajo final para la aprobación del seminario cursado en Mayo). En la Tabla 3 se puede ver cómo se desarrollaron las iteraciones durante este año.



Tabla 3: Iteraciones planificadas durante el año 2020.

Iteración	2020					
	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO
F3I6	■					
F3I7		■				
F3I8		■				
F3I9		■				
F3I10			■			
F4I11					■	
F4I12					■	
F4I13						■
Días	9	9	10	10	9	9
						11

El seguimiento de los planes se realizó durante todo el proceso, debido a que resulta una actividad fundamental en el proceso de la planificación porque, gracias al 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 todo el desarrollo del proyecto. Tenemos que reconocer que las estimaciones realizadas fueron pocas, y que si bien nos ayudaron conocer el tiempo necesario aproximado para la realización de este proyecto, siempre hay que tener en consideración 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 y el seguimiento de las iteraciones siempre se llevó a cabo. Hay que destacar que la duración de las iteraciones planificadas fue evolucionando: en un principio (durante 2017) las planificaciones que se realizaron fueron algo caóticas debido a la cantidad de tareas abarcadas, a la complejidad de las mismas y a la poca experiencia del equipo; pero, a medida que se iba avanzando en el desarrollo, planificar y realizar un seguimiento sobre las tareas planificadas resultaba cada vez más fácil para los miembros del equipo.

Respecto a la fecha límite de entrega del proyecto determinada por el vencimiento de la regularidad de la materia, no hubieron problemas. La fecha límite de entrega del proyecto era finalizados los dos años y medio de regularidad de la materia Laboratorio de Desarrollo de Software: En nuestro caso, ese plazo terminaba en Julio del 2020, pero debido al contexto de pandemia por el 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 involucra las tareas de análisis, priorización y determinación de estrategias para minimizar o reducir el impacto que estos puedan llegar a tener si suceden. Los riesgos son cualquier suceso que de llegar a suceder pueden afectar negativamente a diferentes aspectos del desarrollo del proyecto y pueden impactar en la planificación del mismo. 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 nos permite 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 los mismos. 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, pero sólo en las 3 primeras se seleccionaron riesgos para gestionar. En las siguientes 3 sólo se reducía la cantidad de riesgos a gestionar o se mantenías la gestión de los riesgos ya seleccionados. Los riesgos gestionados, sus categorías y estrategias de mitigación se pueden apreciar en la Tabla 4.

Tabla 4: Riesgos gestionados en el proyecto.

1ra estimación y evaluación de riesgos			
Categoría	ID	Riesgo	Estrategia
Experiencia y capacidad	R05	El grupo de desarrollo no tiene experiencia en cuanto a la conducción de un proyecto de software.	Estrategia de reducción: 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	No se conoce el/los lenguaje(s) de programación a utilizar ni como trabajan.	Estrategia de reducción: 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.
	R09	No se tiene conocimiento ni experiencia respecto al funcionamiento del API de Google Forms.	Estrategia de eliminación: se definió buscar información a cerca de la existencia o no de esa posible API, preguntando en foros de desarrolladores y investigando al respecto.
2da estimación y evaluación de riesgos			
Cronograma	R04	Puede que el límite de tiempo establecido no sea suficiente para desarrollar el proyecto	Estrategia de reducción: 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. Estrategia de contingencia: 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.
Definición del cliente	R02	El cliente puede proponer grandes cambios para el proyecto.	Estrategia de reducción: 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. Estrategia de contingencia: 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.
3ra estimación y evaluación de riesgos			
Complejidad	R16	El sistema de validación de permisos de usuario puede ser complejo de implementar o incluso puede no ser posible implementarlo.	Estrategia de eliminación: se definió programar una reunión con los clientes para explicarles las dificultades encontradas y presentarles posibles alternativas para abordar el problema.
Compromiso del cliente	R01	El cliente no estará presente durante todo el proceso de desarrollo	Estrategia de reducción: 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.

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 se consideraron 15. De esos riesgos considerados se eligió hacer el seguimiento y proponer estrategias para 3 de ellos como se puede ver en la Tabla 4 son los riesgos R05, R07 y R09. Se seleccionaron estos riesgos debido a que eran, según nuestro análisis, los que tenían una probabilidad muy próxima de suceder y por ende impactar en nuestro proyecto.

En términos más generales, la distribución de todos los riesgos identificados en esta primera estimación se pueden ver en la Figura 3 distinguidos entre las diferentes categorías. En esta primera estimación la mayoría de los riesgos identificados se encontraban en las categorías de Experiencia y capacidad, Tecnología y Definición del Cliente. Esta distribución de los riesgos tiene mucho sentido, ya que el equipo se encontraba iniciando el proceso de

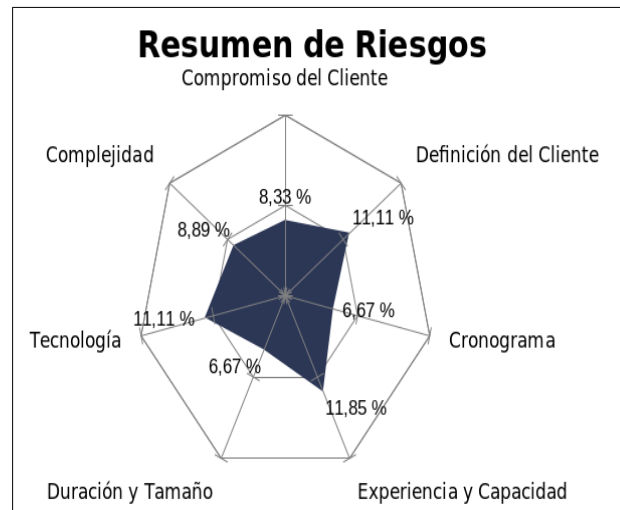


Figura 3: Distribución de los riesgos identificados en la diferentes categorías (I).



Grupo Paire

Ariel Machini y Cinthia Lima

desarrollo, conociendo las tecnologías que teníamos que usar y conociendo los requerimientos del sistema a desarrollar, entonces existían muchos hechos impredecibles asociados con esas actividades y que logramos identificar.

Hay que aclarar que el Resumen de Riesgos se realiza en base a los riesgos identificados y no en base a los considerados.

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, eliminarlos o porque cambiaron los parámetros de análisis del equipo de desarrollo. También aparecieron nuevos riesgos y cambiaron algunas prioridades. En esta segunda evaluación se consideraron 7 riesgos, se lograron mitigar los riesgos R09, R05 y R07 y se decidió sumar a la gestión de riesgos a R04 y R02. En la Tabla 4 se pueden ver estos nuevos riesgos a gestionar junto con sus estrategias de mitigación.

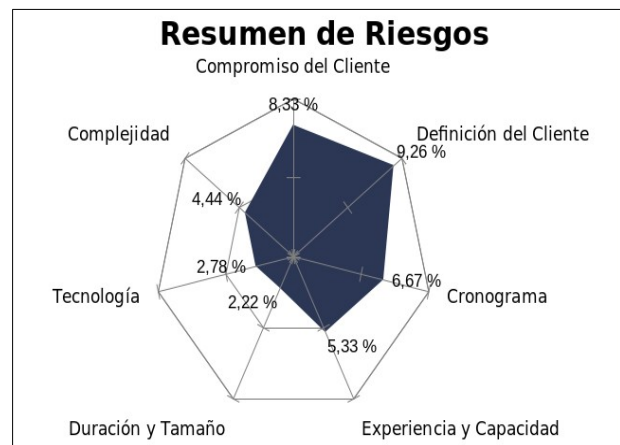


Figura 4: Distribución de los riesgos identificados en las categorías (II).

En la Figura 4 se puede ver como en esta segunda estimación el peso de los riesgos identificados se inclinaban más 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 estábamos diseñando e implementando dichas características pero con la incertidumbre de que los clientes solicitaran algún cambio en los requerimientos muy diferente a lo que se estaba diseñando e implementando.

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. Se consideraron en total 9 riesgos, de los cuales algunos estaban siendo gestionados desde estimaciones anteriores, por otro lado se sumaron a esa gestión de riesgos a R16 y R01, como se puede apreciar en la Tabla 4.

En la Figura 11 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 se desarrolla este trabajo. Por otro lado en la categoría Complejidad apareció un nuevo riesgo crítico (R16) y no plantear este riesgo a los clientes y proponer alternativas para solucionar dicho problema podría atrasar el desarrollo de la aplicación móvil.

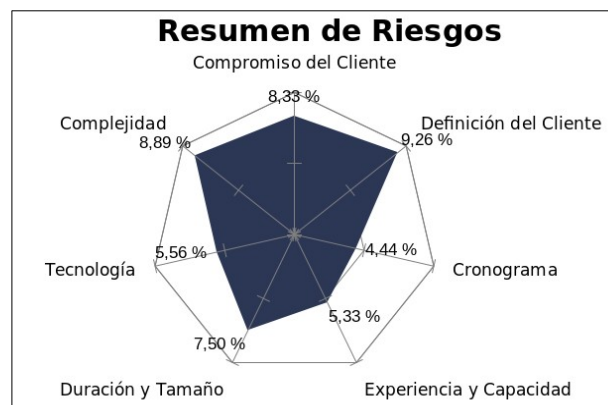


Figura 5: Distribución de los riesgos identificados en las categorías (III).

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 a sólo considerar 3 riesgos. Varios riesgos fueron mitigados y también bajo la probabilidad de ocurrencia de muchos de ellos, por eso es que en esta nueva evaluación la cantidad de riesgos bajo drásticamente.

De los 3 riesgos, dos ya se encontraban siendo gestionados (R01 y R04) y con un seguimiento activo. De los riesgos gestionados hasta esta evaluación se lograron mitigar los siguientes: R02 y 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

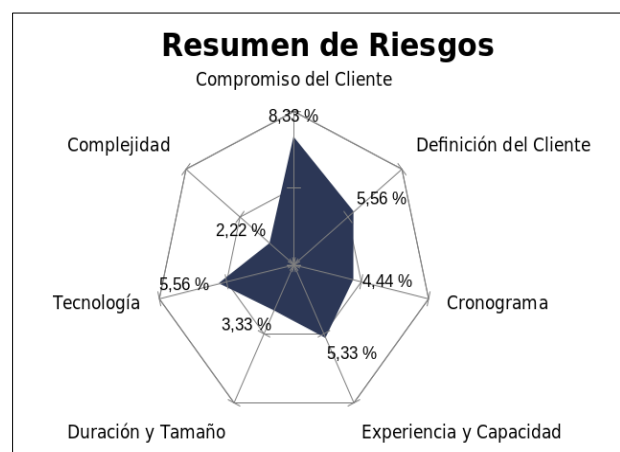


Figura 6: Distribución de los riesgos identificados en las categorías (IV).

clientes una solución para el riesgo R16 de la estimación anterior y por ende ese riesgo tiene un 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

En esta quinta estimación no se identificó ningún nuevo riesgo, y los riesgos a considerar disminuyó a 2: R04 y R01. Ambos riesgos ya se encontraban en seguimiento y se estaban aplicando las estrategias de reducción.

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 ya no fue considerado. Este riesgo está relacionado con la experiencia del equipo para el desarrollo de software, y se mantuvo en consideración en todas las estimaciones pero no llegó a ser considerado nuevamente en esta. Esto se debe a que el equipo estimó que obtuvo experiencia del proceso luego de realizar varias iteraciones y tener un estado del proyecto bastante prometedor para su finalización.

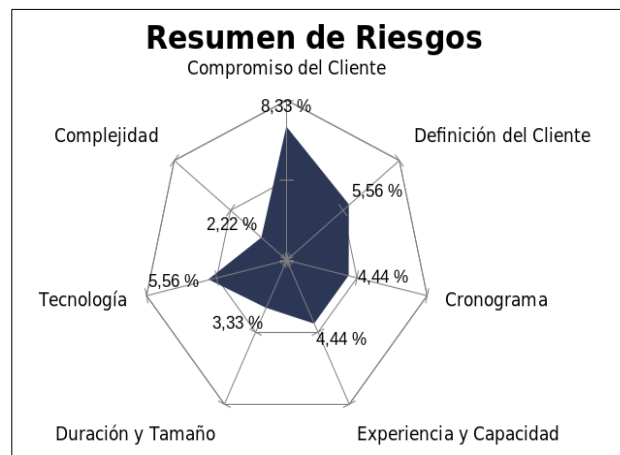


Figura 7: Distribución de los riesgos identificados en las categorías (V).

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 R04 (relacionado con los límites de entrega), ya no se siguió gestionando 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 en Marzo de 2021 y considerando los avances obtenidos era altamente probable que el proyecto este terminado antes de esa fecha. Es por esta razón también que en la Figura 8 la categoría cronograma se redujo a 0%.

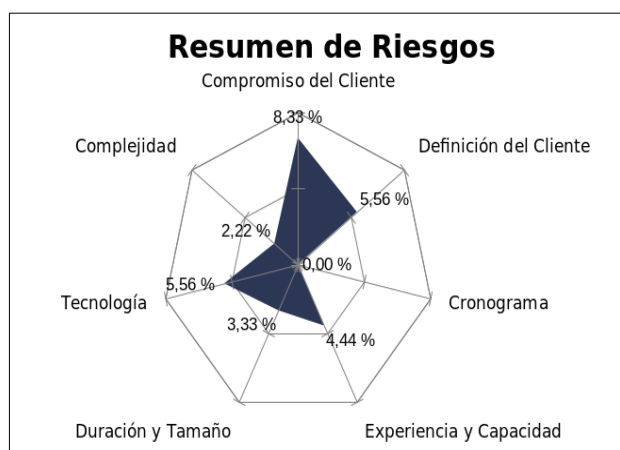


Figura 8: Distribución de los riesgos identificados en las categorías (VI).



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 conveniente hacer estimaciones en un tiempo predefinido. Como mencionamos anteriormente, en otra actividad de aseguramiento de la calidad, la cantidad de estimaciones también depende de las características del proyecto como 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 aspectos del sistema que se iban a probar y se definieron los distintos casos de prueba. El plan de pruebas sólo especificaba las pruebas para la aplicación web y no para la aplicación móvil.

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 planificación de las pruebas tanto para la aplicación web como para la móvil. Si bien en un principio se redactó un plan de pruebas, podemos decir que desde noviembre del 2017 se dejaron de incluir, en las planificaciones de cada iteración, los casos prueba cuando se implementaban módulos del sistema. Planificar y 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.



En la última iteración de construcción se redefinieron casos de prueba para la aplicación web y se definieron para la aplicación móvil ya que nunca se habían definido. Estos casos fueron ejecutados formalmente y el resultado de tal ejecución fue “positivo”. El resultado fue así porque no se encontraron errores ya que se habían solucionado mediante las pruebas informales. En otras palabras, los casos de prueba no cumplieron su función: la de encontrar errores pero de todas maneras su ejecución nos permitió verificar que el sistema funcionaba correctamente.

El principal error cometido en la validación y verificación del software fue no tomar una decisión firme frente a la planificación y ejecución de las pruebas. No existió una fuerte convicción por parte del equipo de desarrollo de seguir incluyendo en cada iteración las pruebas necesarias y esto se debe a qué estábamos más concentrados en las demás actividades del proceso de desarrollo. En este caso, no planificar las pruebas y ejecutarlas de manera continua en el desarrollo no generó grandes inconvenientes, pero en otro tipo de proyecto podría generarlos. Realizar pruebas recién al final del desarrollo podría dejar en evidencia errores que quizás estuvieron presentes desde el inicio de la implementación del sistema, los cuales podrían ser difíciles de resolver por el hecho de no conocer su origen o por requerir mucho tiempo para solucionarlos, retrasando los tiempos de entrega. Dicho suceso se podría evitar si se hubieran realizado pruebas con anterioridad a medida que se implementaban los módulos, ya que al encontrar errores y solucionarlos, se evitaría la generación de un efecto cadena en donde todo o varios módulos del sistema estén comprometidos.

Tomares decisiones más firmes en cuanto a la planificación de las pruebas para futuros proyectos.

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 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.



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de Desarrollo de Software

- **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.

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.

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



Grupo Paire

Ariel Machini y Cinthia Lima

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.
- **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



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de Desarrollo de Software

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.
- **MySQL** (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): 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 de la memoria se detallarán algunas actividades, acontecimientos importantes o problemas que se dieron durante el transcurso de las cuatro fases del proceso de desarrollo del proyecto Colibrí. Cada una de las fases estuvo compuesta por un conjunto de iteraciones, las cuales se detallarán en las siguientes subsecciones según se considere necesario.

Luego de explicar cada una de las fases del proceso, se hará un análisis general de todas las iteraciones en términos de duración (días), cantidad de actividades planificadas y cantidad de actividades completadas a tiempo.

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.

Como se explicó previamente en este documento, 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 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 UARGFlow² (a UARGFlow BS) y la implementación de las aplicaciones del 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

² 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 aplicación web).



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 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 de trabajo.

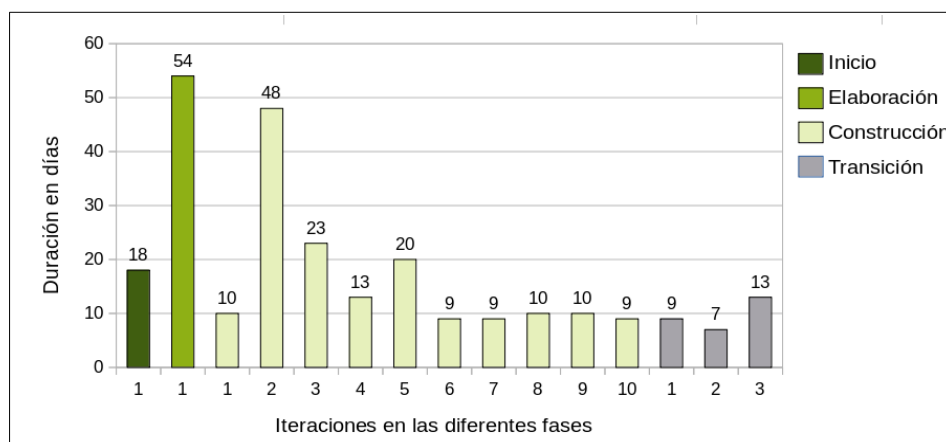


Figura 9: Duración de todas las iteraciones agrupadas por fase.

En la Figura 9 se presenta un gráfico de barras comparativo con la cantidad de días empleados en cada iteración, en el que también se puede distinguir (por color) a qué fase pertenece cada iteración. La iteración que duró más días (54) fue la primera y única iteración de la fase de elaboración: Debido a la poca experiencia del equipo, la duración de esta iteración y cantidad de tareas que abarcó fue muy elevada. Algo similar sucedió en la segunda iteración de la fase de construcción, la cual duró 48 días. Luego de esta iteración, la duración de las iteraciones se comenzó a normalizar gracias al consejo de uno de los profesores (como se explicó con anterioridad). Expresado de otra manera, la duración de las iteraciones fue evolucionando en función del estado del proyecto y de las necesidades de los clientes: Al comienzo, la duración era algo arbitrario y, en consecuencia, las iteraciones planificadas no resultaban eficientes. Con el paso del tiempo, la duración de las iteraciones se fue regulando a medida que se iba conociendo mejor el ritmo de trabajo del equipo. Gracias a esto, las iteraciones resultaron ser más eficientes y productivas.

En la Figura 10 se puede observar la cantidad de tareas definidas en cada iteración en contraste con la cantidad que pudieron ser completadas a tiempo. Contemplando el gráfico, se puede notar que la iteración que comprendió la mayor cantidad de tareas fue la primera iteración de la fase de elaboración. En esta iteración se abarcó un gran número de tareas debido a que quedaron varias tareas pendientes de la iteración anterior. En el gráfico también se puede notar que, en las primeras cuatro iteraciones, las tareas planificadas nunca lograron ser completas a tiempo en su totalidad. Recién a partir de la segunda iteración de la fase de elaboración se puede distinguir un cambio drástico en la cantidad de tareas cumplidas así como también una reducción en la cantidad de tareas planificadas.

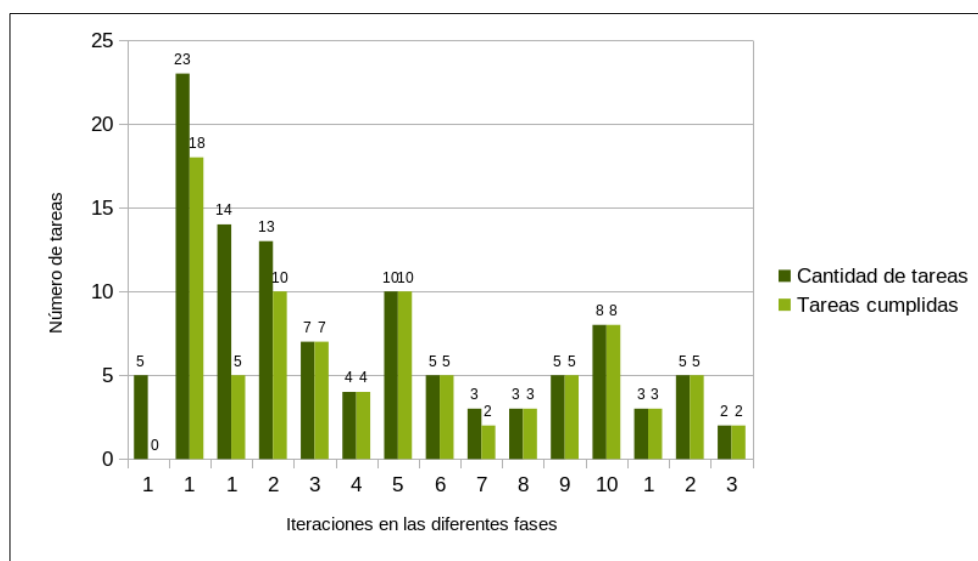


Figura 10: Tareas planificadas VS tareas cumplidas.

En cuanto a la efectividad a la hora de cumplir con el plan de iteración; teniendo en cuenta las 15 iteraciones pertenecientes a las diferentes fases, sólo existieron cinco iteraciones en las cuales no se pudo cumplir a tiempo con las tareas planificadas en el plan de la iteración. En la Figura 11 se puede ver cuáles fueron dichas iteraciones, que representan el 33% de todas las iteraciones planificadas. Por otro lado, las iteraciones en las que se cumplió a tiempo con menos del 50% de las tareas planificadas fueron dos: La primera de la fase de inicio y la primera de la fase de construcción.

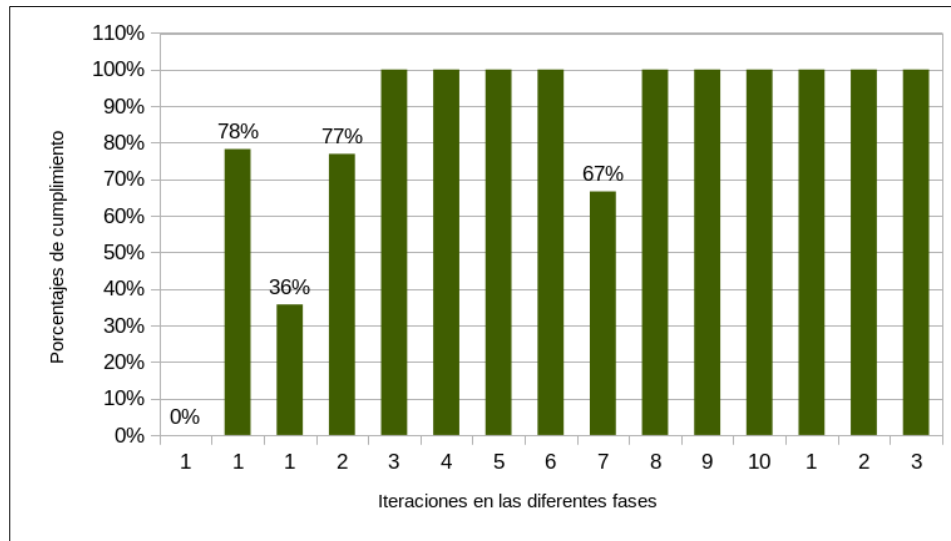


Figura 11: Porcentaje de cumplimiento (a tiempo) de las actividades propuestas en los planes de iteración.

En la primera iteración de la fase de inicio lamentablemente no se logró finalizar con ninguna de las tareas propuestas. Sin embargo, esto no significa que no se hizo nada en la iteración: Al momento de cerrar la iteración, los entregables estaban bastante avanzados (pero no completos).

En la primera iteración de la fase de elaboración se pudo cumplir a tiempo con el 36% de las tareas planificadas. Hay que destacar que en esta fase se dejaron muchos pendientes porque para la regularización de la materia se solicitaron entregables y funcionalidades específicas, lo cual hizo que el equipo se avocara exclusivamente a trabajar sobre dichas tareas. El no cumplimiento de las actividades de la primera iteración acarreó complicaciones a las siguientes iteraciones donde, además de tener que planificar nuevas actividades, también se debían terminar las de fases anteriores. Esto se pudo ver en las primeras cuatro iteraciones (hablando general, sin distinguir fases) que no logran finalizar ese grupo de artefactos incompletos, los cuales recién se pudieron terminar en la tercera iteración de la fase de construcción.

Otra iteración a destacar en la Figura 11 es la séptima de la fase de construcción, en la cual sólo se logró completar a tiempo el 67% de las actividades planificadas. Esto se debió a que no se logró terminar por completo con la implementación de un caso de uso de la aplicación móvil por falta de tiempo y porque surgió un problema inesperado con el procesamiento de los campos de formularios en dicha aplicación.

Conclusión

Como explicamos a lo largo de esta memoria, el proceso de desarrollo del proyecto Colibrí implicó varias tareas en las diferentes disciplinas de la ingeniería del software. Redactar esta memoria nos permitió dejar en evidencia todo el tiempo y el esfuerzo que dedicamos al desarrollo de este producto de software: las actividades que tuvimos realizar para construirlo, las dificultades con las que nos encontramos y cómo los afrontamos.

No cabe duda que gracias al desarrollo de este proyecto pudimos tener nuestro primer acercamiento con el desarrollo de software de alta calidad. A lo largo de este, pudimos aplicar todo lo que aprendimos en las diferentes materias de la carrera: consideramos satisfactorio poder haber empleado de manera integral todo el conocimiento adquirido durante años de cursada en una práctica real. Esta primera experiencia nos dejó varias enseñanzas así como también nos permitió conocer nuestras capacidades individuales y como equipo. Con respecto a lo aprendido, pudimos comprender con mayor claridad las tareas relacionadas con el desarrollo de software, desde el análisis del problema hasta la implementación de una solución que lo resuelva; También aprendimos que la gestión del proyecto es una tarea muy importante y a la vez demandante, pero que es necesaria para poder organizar las tareas, actividad fundamental en el desarrollo de cualquier tipo de proyecto, más aún en el desarrollo de software, ya que por lo general es una actividad iterativa que requiere supervisión constante para conocer el estado del proyecto en cada momento y no perder el control sobre el 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 habrían desarrollado en diferente orden. Pudimos notar que también la gestión de riesgos es importante, ya que sin la detección temprana de riesgos, su seguimiento y la definición de estrategias para su mitigación, el tiempo necesario para el desarrollo hubiese sido más extenso. Esto se puede apreciar en la sección de gestión de riesgos, donde detallamos qué riesgos gestionamos y cuáles estrategias adoptamos.

Por otro lado, las estimaciones y planificaciones que realizamos nos fueron de utilidad para poder progresar con la confianza de que seguíamos el camino correcto y que los tiempos estipulados serían suficientes. Por suerte, y como explicamos en la sección de estimación y planificación, no tuvimos contratiempos con la entrega del producto de software.

En cuanto a la comunicación, existieron dos *relaciones*, presentes durante todo el proceso de desarrollo. La primera, fue la relación entre los miembros del equipo de trabajo, la cual fue muy amena. Coincidimos en que, si bien hubieron largas conversaciones para la toma de ciertas decisiones (sean de planificación, diseño, implementación, etcétera), siempre pudimos llegar a un



Grupo Paire

Ariel Machini y Cinthia Lima

acuerdo. En gran parte de los casos, las decisiones a tomar tuvieron que ver con preferencias particulares que, obviamente, cada persona tiene y no siempre coinciden; es por esta razón que es importante que, al surgir estas diferencias, el equipo tenga la capacidad para poder afrontarlas y resolverlas, para así poder tomar la decisión que sea mejor para el proyecto y para el cliente.

La segunda relación fue la que mantuvimos con los clientes, quienes en este caso también fueron los profesores de la materia en la cual desarrollamos este proyecto. Este fue nuestro primer acercamiento a lo que sería un *cliente real*, y gracias a ello finalmente pudimos hacer una toma de requerimientos real, luego de haber practicado dicha actividad a lo largo de tantas materias. Pudimos apreciar lo importante que es mantener una buena comunicación con el cliente; el tenerla nos facilitó mucho seguir el nivel de satisfacción del mismo, así como también nos permitió poder avanzar con seguridad en las tareas del desarrollo. Gracias a la comunicación constante que mantuvimos con los clientes, pudimos conocer sus impresiones acerca del trabajo que íbamos realizando y sus cambios en los requerimientos.

El desarrollo de este proyecto también nos permitió aprender nuevos lenguajes y tecnologías que no conocíamos o no utilizábamos. También, gracias a esta oportunidad, descubrimos que nuestra curva de aprendizaje fue relativamente baja debido a toda la práctica que adquirimos en las materias del área de programación de la carrera. El aprendizaje de estas nuevas tecnologías nos capacitó para posibles futuros desarrollos en los que sea un requerimiento conocer dichas tecnologías.

Con respecto a las pruebas, reconocemos la falta de convicción que tuvimos en la planificación de la ejecución de las mismas. Efectivamente diseñamos casos de prueba (para la aplicación web), sin embargo no planificamos correctamente su ejecución durante el proceso de desarrollo. En el caso de la aplicación móvil ocurrió algo similar: Recién definimos casos de prueba (y su ejecución) tras terminar con su codificación. Entonces, si bien realizamos pruebas y corregimos errores informalmente durante todo el desarrollo y ejecutamos un lote de pruebas al final del mismo, la planificación de la ejecución de pruebas no existió durante el transcurso de toda la fase de construcción del proceso de desarrollo. A raíz de esto, no se documentaron los errores detectados, el esfuerzo que fue necesario para resolverlos y el tiempo tiempo que nos llevó resolverlos. Nos aseguraremos de no cometer el mismo error (la falta de planificación y de una verificación formal del correcto funcionamiento del sistema) en futuros desarrollos, ya que puede ser la causa de problemas en el desarrollo del sistema.

También consideramos que es importante repasar los problemas que tuvimos en relación con la planificación de tareas en la primera parte del proceso de



desarrollo: Los plazos que definíamos no eran adecuados y/o planificábamos demasiadas actividades, cosas que resultaban en iteraciones ineficientes y actividades no terminadas a tiempo. Tras unas cuantas iteraciones mal planificadas, y a raíz de una recomendación de uno de los profesores de la materia, pudimos definir un ritmo y un nivel de carga para las iteraciones con el que nos sentíamos cómodos y que mantuvimos durante el resto del proceso de desarrollo; gracias a estos cambios logramos realizar iteraciones más productivas, cosa que explicamos más en detalle al final de la sección de desarrollo del proyecto. Debemos decir también que la experiencia que obteníamos a medida que íbamos ejecutando cada iteración nos permitió llevar las tareas de gestión del proyecto con cada vez más naturalidad. En relación a esto, queremos destacar y agradecer el constante acompañamiento que tuvimos por parte de los docentes de la materia, que guiaron durante todo el proceso de desarrollo. Más allá del seguimiento de la metodología PSI, que también nos fue de mucha ayuda ya que brindaba mucha información sobre el proceso y sobre los entregables que forman parte del producto de software, la participación, el apoyo y la motivación brindada por los docentes fue fundamental para la realización y correcta culminación de este proyecto.

En cuanto al producto de software que desarrollamos (el sistema “Colibrí”), debemos decir que estamos muy contentos de que se nos haya asignado la resolución de esta problemática en la materia Laboratorio de Desarrollo de Software. Lo que empezó como un pequeño proyecto para digitalizar los formularios del SIT (Servicio de Informática y Telecomunicaciones) en la universidad con el tiempo (y con diversos cambios en los requerimientos) se convirtió en un complejo proyecto que permitirá mejorar los distintos procesos de la universidad mediante la digitalización de formularios habidos y por haber. Ver que pudimos desarrollar un sistema que será de utilidad para la universidad nos alegra; y debemos decir que nos esforzamos mucho para poder entregar un sistema completamente funcional, libre de errores, fácil de utilizar y mantener. Consideramos que el producto está listo para usar pero puede requerir una pequeña capacitación para la funcionalidad que permite crear formularios, ya que la consideramos una de las más extensa.

En conclusión, desarrollar el proyecto Colibrí nos dejó experimentar el proceso de desarrollo de software en su totalidad y, gracias a esto, pudimos crear un producto de software para responder a un problema real. Todo lo que aprendimos durante el desarrollo de este proceso nos será de utilidad en futuros proyectos.



Grupo Paire

Ariel Machini y Cinthia Lima

Laboratorio de Desarrollo de Software