

# Memoria del Proyecto de Software Testify

## OSLO

Ojeda Valeria – Sly Eduardo  
Levipichun Emilio – Oyarzo Malena





Memoria  
Testify

## Contenido

<b>Objetivos</b>	<b>4</b>
<i>Objetivo General</i>	4
<i>Objetivos Específicos</i>	4
<b>Capítulo I: Aspectos clave del proyecto</b>	<b>5</b>
<i>Metodología de Desarrollo</i>	5
<i>Estructura del trabajo en equipo</i>	6
<i>Gestión del Proyecto y Calidad</i>	9
<i>Planificación del Proyecto</i>	11
<i>Estimación</i>	12
<i>Gestión de Riesgos</i>	14
<i>Pruebas de Validación y Verificación</i>	18
<i>Gestión SQA para el aseguramiento de la calidad del software</i>	21
<i>Tecnologías Utilizadas</i>	26
<b>Capítulo II: Fases del Proyecto</b>	<b>28</b>
<i>Inicio</i>	28
<i>Elaboración</i>	29
<i>Inicio de la Elaboración – Iteración E01 (09/09 al 23/09)</i>	29
<i>Inicio de la Elaboración – Iteración E02 (23/09 al 07/10)</i>	33
<i>Construcción</i>	36
<i>Fase de Construcción 1</i>	36
<i>Fase Construcción 2</i>	38
<i>Fase Construcción 3</i>	38
<i>Conclusión Fase Construcción</i>	39
<i>Transición</i>	40
<b>Conclusiones</b>	<b>42</b>
<i>Conclusión individual Malena Oyarzo</i>	42
<i>Conclusión individual Emilio Levipichun</i>	43
<i>Conclusión Individual Valeria Ojeda Muñoz</i>	44
<i>Conclusión Final</i>	45

# Memoria Proyecto Testify

---

El presente documento constituye la Memoria del Proyecto del Sistema de Gestión de Pruebas de Software, denominado **Testify**, desarrollado en el marco de la asignatura Laboratorio de Desarrollo de Software, correspondiente al tercer año de las carreras de Analista de Sistemas y Licenciatura en Sistemas, dictadas en la Unidad Académica Río Gallegos de la Universidad Nacional de la Patagonia Austral.

Dicha asignatura tiene como objetivo principal que los estudiantes participen en un entorno de desarrollo de software que les permita identificar cada etapa del proceso e implementar las técnicas y metodologías aprendidas mediante un enfoque integrador.

A lo largo del documento se detallan los aspectos más relevantes del proyecto, incluyendo la organización del equipo, las expectativas tanto del equipo docente como del estudiantado, los problemas identificados y sus respectivas soluciones, las metodologías de trabajo adoptadas por el equipo OSLO, la planificación realizada, las herramientas y tecnologías utilizadas, así como las conclusiones generales del proyecto y las reflexiones personales de cada integrante.

## Objetivos

### Objetivo General

Desarrollar el proyecto de un Sistema de Gestión de Pruebas de Software, denominado Testify, destinado a su implementación en el marco de la asignatura Laboratorio de Desarrollo de Software, correspondiente a las carreras de Analista de Sistemas y Licenciatura en Sistemas.

### Objetivos Específicos

- Coordinar el trabajo en equipo para el desarrollo de la propuesta presentada por el equipo de cátedra para ejecutar durante la cursada de la asignatura Laboratorio de Desarrollo de Software.

- Cumplir con todas las pautas establecidas en cada una de las etapas del desarrollo para llegar a la regularización de la materia.
- Presentar el Proyecto finalizado antes que se cumpla el plazo de regularidad de la asignatura.

## Capítulo I: Aspectos clave del proyecto

### Metodología de Desarrollo

El desarrollo del sistema Testify se organizó siguiendo la Metodología PSI. Este proceso se basa en los principios del Proceso Unificado para el Desarrollo de Software (RUP), pero adaptado a las necesidades del caso. PSI no es una metodología cerrada, sino un marco de trabajo extensible, que puede ajustarse según las características específicas de cada organización o proyecto.

El proceso PSI se estructura en cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase puede contener una o más iteraciones, dependiendo de la complejidad y del tamaño del proyecto. Cada iteración sigue una lógica iterativa e incremental, donde se trabaja con un subconjunto del sistema que se analiza, diseña, implementa y prueba, generando un incremento funcional del producto.

Una de las características distintivas de PSI es que es dirigido por casos de uso. Esto significa que los requerimientos funcionales del sistema se capturan a través de casos de uso, los cuales se utilizan como eje central para definir el contenido y las prioridades de cada iteración. Durante el desarrollo de Testify, seleccionamos y organizamos los casos de uso de forma tal que en cada iteración pudiéramos abordar aquellos que permitieran avanzar progresivamente en la construcción de las funcionalidades clave.

Otra particularidad del proceso es que está centrado en la arquitectura. Se reconoce que ningún modelo único puede capturar todos los aspectos de un sistema complejo,

por lo que se trabaja con múltiples modelos y vistas, incluyendo el modelo de casos de uso, el modelo de clases, el modelo de datos y el prototipo funcional. Durante la fase de elaboración del proyecto, uno de nuestros objetivos principales fue justamente desarrollar una línea base arquitectónica sólida, que sirviera de sustento para el resto de las fases.

El PSI también está enfocado en los riesgos. Desde las primeras fases del proyecto, se espera que el equipo identifique, documente y mitigue los riesgos críticos. En nuestro caso, trabajamos activamente en la gestión de riesgos mediante la elaboración de planes específicos y su monitoreo en cada iteración, especialmente durante la fase de Elaboración, donde varios de los riesgos detectados fueron cerrados o tratados con acciones concretas.

Cada iteración dentro del PSI aborda distintas disciplinas: análisis de requerimientos, diseño, implementación y prueba. Aunque todas las disciplinas están presentes en cada fase, la proporción del esfuerzo dedicado a cada una varía según el momento del proyecto. Por ejemplo, en la fase de Inicio predominan actividades relacionadas con la definición de requerimientos y el modelado del negocio, mientras que en la fase de Construcción se incrementa el trabajo en implementación y pruebas.

Por lo tanto, PSI nos permitió trabajar con una metodología estructurada, adaptable y orientada a resultados concretos, promoviendo la entrega de productos funcionales en cada iteración, con foco en la arquitectura, la calidad y el control de riesgos.

## **Estructura del trabajo en equipo**

Desde el inicio del segundo cuatrimestre del año 2024, el equipo OSLO, conformado por Valeria Ojeda, Eduardo Sly, Emilio Levipichun y Malena Oyarzo, asumió el desafío de desarrollar el sistema Testify como parte de la asignatura Laboratorio de Desarrollo de Software. El nombre del equipo surge de las iniciales de los apellidos de sus integrantes

y simboliza la colaboración y responsabilidad compartida que caracterizó al grupo durante todo el proceso de trabajo.

Uno de los aspectos fundamentales en la organización del equipo fue la evaluación inicial de fortalezas y debilidades de cada integrante. Esta autoevaluación permitió detectar las habilidades técnicas y blandas de cada miembro, así como también aquellas áreas que requerían acompañamiento o refuerzo. A partir de este diagnóstico, se definieron los roles principales y las tareas a asumir por cada integrante, de forma tal que cada uno pudiera aportar lo mejor de sí al proyecto, sin dejar de lado el aprendizaje colaborativo y el crecimiento mutuo. Los roles definidos fueron los siguientes:

- Valeria Ojeda asumió el rol de Líder de Proyecto, coordinando el grupo y gestionando los plazos, entregables y comunicación con los docentes. También participó como Analista, Documentadora, Tester y Responsable de Validación, aprovechando su capacidad organizativa, visión integral del proyecto y compromiso con la calidad.
- Emilio Levipichun se desempeñó como Administrador de Configuración y Administrador SQA, siendo responsable del control de versiones, la trazabilidad de los entregables y el seguimiento de la calidad. También participó activamente en la documentación del sistema y en la gestión de la configuración, asegurando la correcta administración de los artefactos del proyecto, su labor se destacó por su constancia y responsabilidad técnica en la mejora de los procesos de calidad.
- Eduardo Sly se encargó principalmente de la Arquitectura y Diseño del sistema, así como del desarrollo del código fuente en los lenguajes seleccionados. Su experiencia previa en programación y pensamiento analítico fueron claves para el diseño robusto y escalable de la solución.
- Malena Oyarzo, por su parte, fue la principal Documentadora, participó como Analista y asumió también el rol de Tester, garantizando la trazabilidad de los requerimientos y el cumplimiento de los criterios de calidad definidos. Su minuciosidad en la redacción y análisis resultó esencial en la construcción de los

entregables documentales y en la verificación del cumplimiento funcional del sistema.

La estructura organizacional del equipo combinó una jerarquía funcional con una colaboración horizontal. Aunque los roles estaban claramente definidos, se promovió un modelo de trabajo cooperativo, donde cada integrante podía colaborar o apoyar en otras áreas según las necesidades del proyecto o la disponibilidad de tiempo. Esto permitió una mayor flexibilidad, favoreció el aprendizaje entre pares y mejoró la cohesión del grupo.

Además, se establecieron instancias periódicas de revisión y feedback interno, en las que se discutía el avance de cada línea de trabajo, se compartían dificultades y se evaluaba la necesidad de redistribuir tareas. Este enfoque permitió atender imprevistos, prevenir cuellos de botella y fortalecer el compromiso del equipo con los objetivos comunes.

Para organizar y visualizar las tareas del equipo, se utilizó la herramienta Trello, implementando la metodología Kanban. Esta práctica permitió asignar responsabilidades, gestionar el avance de las actividades y tener una visión clara del estado de cada ítem del proyecto. Las columnas utilizadas (como Por hacer, En progreso y Finalizado) facilitaron el seguimiento individual y colectivo de los compromisos asumidos, fomentando la autogestión y la transparencia en el proceso de desarrollo. Se ampliará esta información en la sección Gestión del Proyecto y Calidad.

En cuanto a la comunicación, el equipo OSLO combinó reuniones virtuales a través de la plataforma Google Meet con reuniones presenciales realizadas en el Campus de la UNPA-UARG, lo que permitió mantener la continuidad del trabajo en equipo, resolver bloqueos en tiempo real y tomar decisiones consensuadas de forma eficiente.

## **Gestión del Proyecto y Calidad**



Para la gestión del desarrollo de Testify, adoptamos un enfoque basado en el Proceso Unificado de Desarrollo de Software (RUP), lo que nos permitió organizar el trabajo de manera iterativa e incremental. Durante el segundo cuatrimestre del 2024, establecimos un ritmo de trabajo regular mediante sprints quincenales, en los que definimos objetivos concretos, distribuimos tareas y evaluamos el avance general del proyecto.

Desde un comienzo, implementamos la herramienta Trello como soporte central para la organización de tareas y la colaboración entre los integrantes del equipo. Esta herramienta nos permitió aplicar la metodología Kanban, representando visualmente el estado de cada actividad dentro de un tablero estructurado en las siguientes columnas: Backlog (tareas próximas), Por hacer, En proceso y Finalizado.

Cada tarjeta de Trello correspondía a una tarea específica e incluía un identificador único bajo el formato TXX IXX, donde "T" indicaba el número de tarea e "I" el número de iteración o sprint. Esto facilitó el seguimiento, la trazabilidad y el control de las actividades planificadas. Dentro de cada tarjeta registramos descripciones detalladas, fechas de inicio y finalización, responsable asignado, etiquetas categorizadoras, archivos adjuntos y listas de subtareas (checklists), lo que nos permitió mantener un orden claro y coherente.

Una ventaja significativa del uso de Trello fue su impacto positivo en nuestras presentaciones ante la cátedra. Gracias a la organización visual y actualizada del tablero, pudimos mostrar claramente qué tareas se habían cumplido y qué decisiones se tomaron durante su ejecución, facilitando la explicación del proceso y las justificaciones técnicas. Además, el historial de tareas descartadas también quedó registrado, lo que nos permitió fundamentar los cambios ante el equipo docente.

Durante la planificación del proyecto, se presentó un caso particular de ausencia programada de uno de los integrantes del equipo. Al haberlo sabido con anticipación, reorganizamos nuestras actividades de forma tal que el resto del grupo pudiera

continuar sin inconvenientes. Esta experiencia destacó la importancia de una planificación realista y la necesidad de contar con herramientas colaborativas como Trello, que nos permitieron mantener el ritmo de trabajo sin perder visibilidad sobre el avance general.

Las reuniones del equipo se realizaron tanto de forma virtual, a través de Google Meet, cómo de manera presencial en el campus de la UNPA-UARG. Estas instancias nos ayudaron a coordinar decisiones, ajustar tareas según los cambios que surgían y evaluar el cumplimiento de los objetivos definidos en cada sprint.

Desde el punto de vista de la calidad, definimos e implementamos un Plan de Aseguramiento de la Calidad (SQA), que estableció estándares, métricas, y procedimientos para asegurar que todos los entregables, tanto de software como documentales, cumplieran con los requerimientos funcionales y no funcionales acordados. Este plan incluyó revisiones técnicas formales, control de la documentación, evaluación de cumplimiento de requerimientos, pruebas de calidad y validaciones internas.

También desarrollamos un Plan de Gestión de Riesgos, que fue revisado y actualizado al inicio de cada sprint. Identificamos riesgos asociados a tiempos, herramientas, experiencia técnica, roles y coordinación, y definimos estrategias de mitigación y planes de contingencia para cada uno. Esta práctica nos permite anticiparnos a posibles obstáculos y mantener la estabilidad del proyecto ante situaciones imprevistas.

A su vez, implementamos un sólido Plan de Gestión de Configuración, basado en el uso de Git y GitHub, con una estrategia clara de ramas y un control riguroso de versiones mediante pull requests y validaciones. Esto aseguró la trazabilidad de los cambios, la integridad de los entregables, y la correcta integración del trabajo de todos los integrantes.

## Planificación del Proyecto

Desde el inicio del segundo cuatrimestre, abordamos la planificación de Testify con una visión realista, estratégica y colaborativa. Partimos de los requerimientos definidos en la fase inicial y, sobre esa base, construimos una hoja de ruta detallada que orientó cada etapa del proyecto, adaptándola a los tiempos, recursos y características del equipo.

Una de las primeras decisiones fue estructurar el trabajo en iteraciones quincenales (sprints), adoptando un enfoque ágil que nos permitiera realizar entregas parciales, obtener retroalimentación constante y aplicar mejoras continuas. Cada sprint contó con una planificación específica, que incluyó objetivos definidos, distribución de tareas, asignación de responsables y fechas de entrega.

Para organizar estas actividades, elaboramos un cronograma general que fuimos ajustando conforme a los avances, imprevistos y decisiones del equipo. Este cronograma estuvo acompañado por entregables formales definidos en el Plan de Proyecto, como la Propuesta de Desarrollo, la Especificación de Requerimientos, los Modelos de Casos de Uso y de Datos, el Prototipo Funcional, y los documentos de diseño y componentes, entre otros.

La utilización de Trello fue fundamental para materializar esta planificación en acciones concretas. A través del tablero, distribuimos las tareas correspondientes a cada sprint, lo que nos permitió mantener en todo momento una visión clara del avance del proyecto. La trazabilidad de cada actividad, junto con el registro de fechas, responsables y comentarios en tiempo real, convirtieron nuestra planificación en una herramienta dinámica, flexible y eficaz.

A lo largo de todo el proceso, realizamos reuniones de planificación y seguimiento, tanto virtuales como presenciales, en las que analizamos el estado de los sprints, evaluamos posibles desvíos respecto de lo planificado y, en caso necesario, ajustamos

los tiempos o redistribuimos las actividades. Esta dinámica nos permitió mantener el control del proyecto sin resignar flexibilidad.

## Estimación

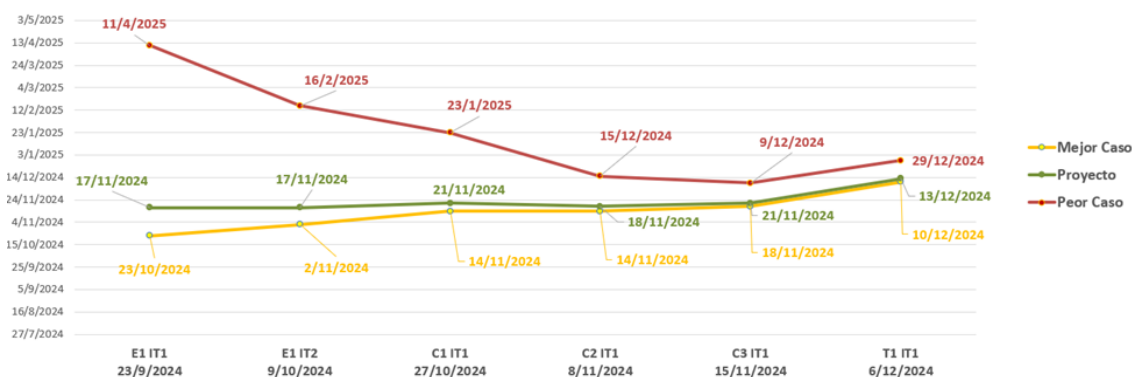
La estimación, tal como se abordó en otras asignaturas de las carreras de Analista y Licenciatura en Sistemas, es un proceso fundamental que permite planificar y administrar eficazmente los recursos, los tiempos y los costos involucrados en el desarrollo de un proyecto de software.

Para el desarrollo del sistema de gestión de pruebas Testify, se utilizó el método de Puntos de Casos de Uso, siguiendo las plantillas provistas en la página de la Metodología de Desarrollo PSI de la UARG-UNPA.

Durante el segundo cuatrimestre, período en el cual se cursó la asignatura, se realizaron las estimaciones correspondientes de manera iterativa, analizando y definiendo los valores de los factores involucrados en dicho método.

A continuación, se presenta un resumen de la evolución del proyecto a partir de las estimaciones calculadas al cierre de cada iteración durante el año 2024. Para ello, se consideraron tres escenarios: mejor caso, peor caso y caso más probable.

La diferenciación entre estos escenarios se realizó a partir de supuestos extremos, asignando valores específicos a los factores involucrados en la estimación. Esto permitió establecer un rango de comparación que se mantuvo a lo largo de toda la cursada.



**Gráfico 1. Evolución de las Estimaciones de Testify**

De acuerdo con el Gráfico 1, pueden realizarse las siguientes observaciones:

- Las estimaciones comienzan a calcularse en la Fase de Elaboración 1, Iteración 1. Durante la Fase de Inicio no fue posible realizarlas, ya que en esa etapa se analizó la propuesta, se conformó el equipo de trabajo OSLO y se establecieron pautas organizativas previas al desarrollo. Resultaba fundamental, en ese momento, definir con precisión los objetivos del proyecto para evitar retrabajos posteriores.
- En las fases de Elaboración se observa que, en general, los tiempos estimados del caso más probable se mantuvieron por encima del mejor caso y relativamente cercanos al peor caso. Esto se debió a que el equipo estaba integrado por cuatro personas, lo que permitió distribuir las tareas de forma eficiente y avanzar tanto en el desarrollo como en la gestión de los casos de uso.
- La evolución del gráfico muestra cómo la curva del caso más probable tiende a converger con la del mejor caso, lo cual indica que las estimaciones reales se fueron ajustando progresivamente a los valores más optimistas.
- Como conclusión, consideramos que el proceso de estimación fue una herramienta muy útil, que supimos aprovechar para organizarnos y cumplir con los plazos establecidos. Además, sirvió para visualizar la evolución del equipo en relación con distintos factores: no solo la resolución de los casos de uso, sino también el tiempo efectivamente dedicado, la experiencia adquirida en el rol de

liderazgo, el dominio del lenguaje de programación y otros aspectos que fueron madurando a lo largo del desarrollo.

## Gestión de Riesgos

La gestión de riesgos es un proceso esencial en el desarrollo de software, ya que permite identificar, evaluar y controlar eventos o situaciones que podrían afectar negativamente el cumplimiento de los objetivos del proyecto.

Para llevar adelante este proceso, el equipo OSLO utilizó una planilla elaborada en Excel, suministrada por la cátedra como parte del conjunto de herramientas disponibles en el sitio web de la Metodología de Desarrollo PSI. Esta planilla permite clasificar los riesgos por categorías, asignarles un valor de impacto y estimar la probabilidad de ocurrencia. A partir de ello, se genera una tabla que calcula el factor de ocurrencia de cada riesgo. Los riesgos se ordenan de mayor a menor según ese factor, lo que permite identificar con claridad cuáles son los más relevantes.

El equipo comenzó con el análisis de riesgos durante la primera iteración de la Fase de Elaboración y continuó realizando un seguimiento en cada cierre de iteración. Este seguimiento incluyó tanto la evaluación de los riesgos inicialmente identificados como la consideración de nuevos riesgos que pudieran surgir durante el desarrollo de Testify.

A continuación, se presentan los gráficos que ilustran el proceso de evaluación, análisis y gestión de riesgos implementado por el equipo.

RIESGO	DESCRIPCIÓN BREVE DEL RIESGO
RK87	Posibles inasistencias de cualquiera de los integrantes del equipo OSLO
RK86	No se detectan todos los errores a tiempo durante las pruebas del software
RK41	Las pruebas sobre los casos de usos llevan más tiempo del planificado
RK60	Cambios inesperados en los objetivos del proyecto o requisitos
RK40	Falta de capacitación de los integrantes del equipo OSLO
RK20	Falta de tiempo por trabajo/carga horaria
RK19	Cortes de electricidad e Internet
RK4	No entender o definir bien los requerimientos funcionales
RK3	No se realizan suficientes pruebas al software

Tabla 1. Riesgos de Testify

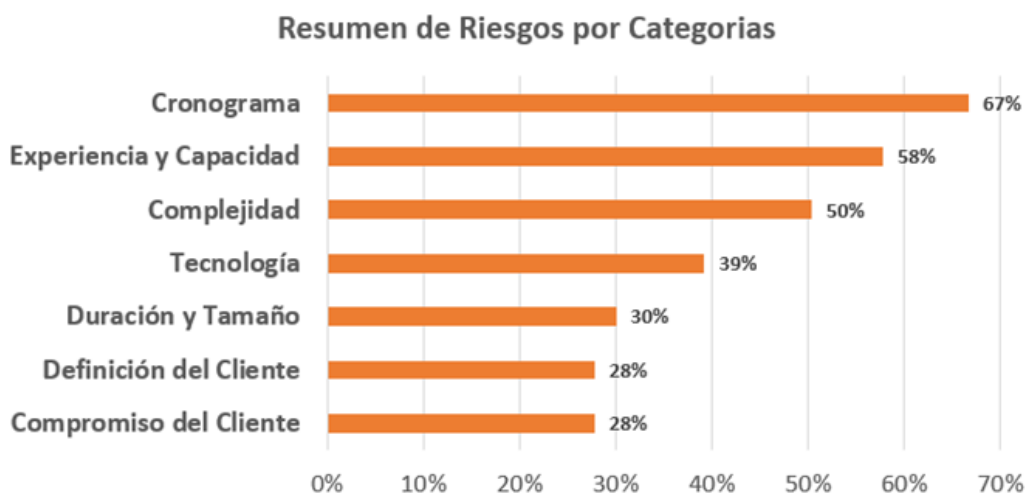
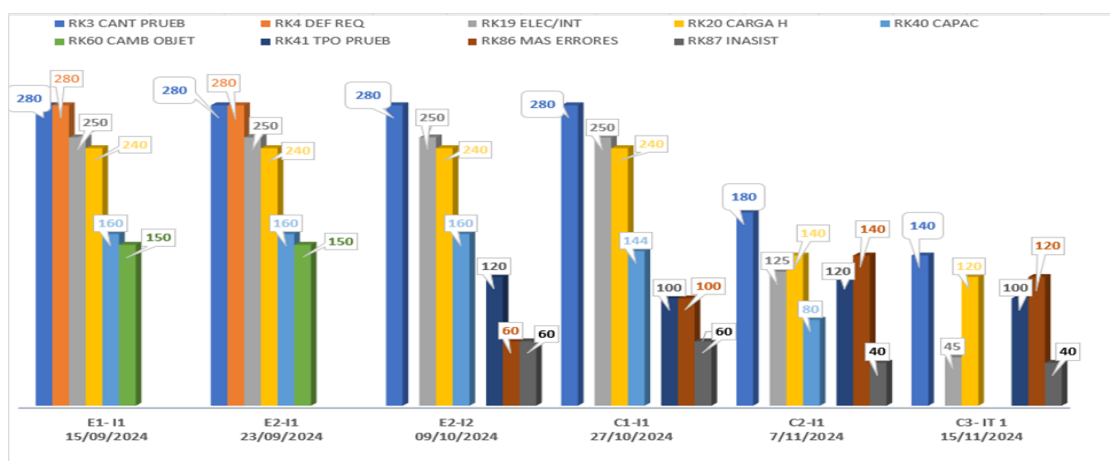


Gráfico 2. Resumen de Riesgos por Categorías

En el Gráfico 2 se representa el porcentaje de probabilidad de ocurrencia de los riesgos, clasificados según su categoría. Se observa que los riesgos vinculados al cronograma presentan el valor más elevado, con un 67 %, seguidos por la categoría experiencia y capacidad, con un 58 %, y complejidad, con un 50 %. Las restantes categorías presentan probabilidades de ocurrencia inferiores al 50 %.



**Gráfico 3. Riesgos por Iteración**

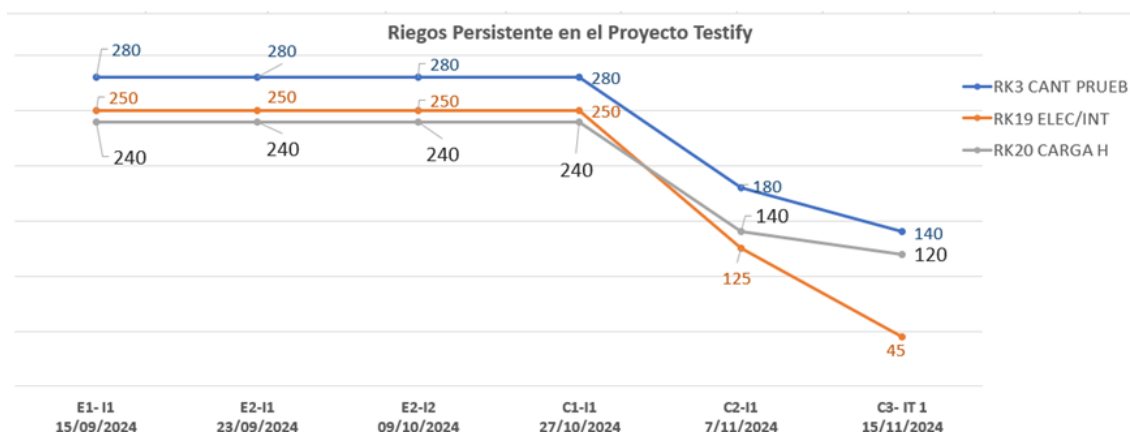
El Gráfico 3 muestra los riesgos con mayor factor de ocurrencia en cada una de las iteraciones realizadas. Cada barra de color representa un riesgo específico.

- En el gráfico se identifican tres riesgos que se mantuvieron activos durante todas las iteraciones del proyecto: RK3, RK19 y RK20.
- Por otra parte, los riesgos RK60 y RK4 estuvieron presentes únicamente en las dos primeras iteraciones de la Fase de Elaboración 1.
- En tanto, los riesgos RK60, RK41 y RK87 comenzaron a registrarse a partir de la segunda iteración de la Fase de Elaboración 2.

A partir de las observaciones realizadas en el Gráfico 3, se concluye que se llevó a cabo un seguimiento constante de los riesgos a lo largo del desarrollo del proyecto. Si bien algunos riesgos fueron cerrados, también surgieron nuevos en distintas fases, lo que evidencia una gestión activa y adaptativa.



Se destaca, además, la evolución positiva en la gestión de riesgos: a medida que avanzaba el proyecto, el factor de ocurrencia de los riesgos activos fue disminuyendo. Esto se logró gracias a la implementación efectiva de planes de mitigación y contingencia, que permitieron reducir su impacto y mejorar la capacidad de respuesta del equipo OSLO.



**Gráfico 4. Riesgos persistentes en Testify**

El Gráfico 4 muestra cómo los tres riesgos que estuvieron presentes a lo largo de todo el desarrollo del proyecto (RK3, RK19 y RK20) fueron disminuyendo su factor de ocurrencia con el paso del tiempo. Esta reducción se debió a distintos factores, que pueden analizarse en relación con cada riesgo.

En las etapas iniciales, los factores de estos riesgos se mantuvieron estables. En el caso de RK3, relacionado con posibles fallos en las pruebas, el riesgo permanecía activo, pero aún no se había iniciado el período de pruebas, por lo que no era posible evaluar su impacto ni aplicar acciones mitigadoras. Respecto a RK20, asociado a la disponibilidad horaria del equipo, su factor fue elevado al comienzo debido a que las primeras fases requerían una carga significativa de trabajo en análisis, diseño y gestión, en un contexto donde todos los integrantes del equipo tenían otras obligaciones laborales. Finalmente, RK19, vinculado a los cortes de energía en la ciudad, también mostró un factor alto al inicio, coincidiendo con un período en el que estos eventos eran frecuentes.

Tras el análisis de la evolución de los riesgos, podemos concluir que el seguimiento sistemático durante el desarrollo del proyecto permitió obtener múltiples beneficios: mejora en la calidad del producto, prevención de problemas futuros, asignación más eficiente de recursos y tareas, y toma de decisiones estratégicas fundamentadas. Estos aspectos contribuyeron significativamente al éxito del proyecto Testify.

## Pruebas de Validación y Verificación

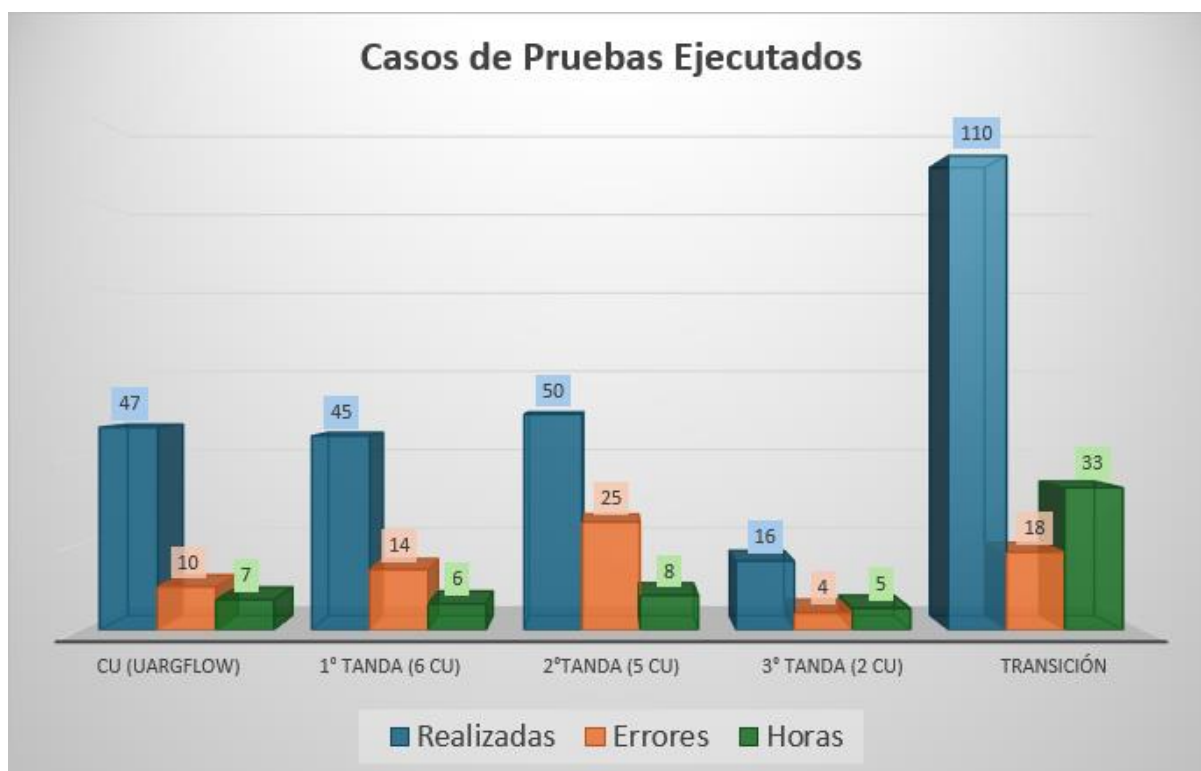
Durante el desarrollo de la aplicación Testify, se llevaron a cabo distintas pruebas y validaciones a lo largo de todas las etapas del modelo RUP. En esta sección se detallan las actividades realizadas en cada fase, con el objetivo de asegurar la calidad del producto final.

- **Inicio:** Durante esta fase se realizaron actividades preliminares para validar la viabilidad del proyecto. Se llevó a cabo la validación de los requisitos iniciales con los clientes mediante dos entrevistas y un estudio del mercado actual. Para ello, se analizaron aplicaciones como TestRail, PractiTest y ALM Octane, las cuales sirvieron como referencia y fuente de ideas para ajustar y completar los requerimientos del sistema.
- **Elaboración:** En esta etapa se profundizó en el diseño del sistema. Se generaron múltiples versiones de prototipos, tanto mockups en Excel como prototipos evolutivos desechables en Figma. Estos fueron utilizados para realizar pruebas al paso, que permitieron obtener retroalimentación temprana de los clientes sobre aspectos clave de la aplicación Testify. Además, se definieron el plan de pruebas a ejecutar, el modelo de datos y la arquitectura del software, con el objetivo de evaluar su rendimiento y escalabilidad.
- **Construcción:** Durante esta fase se iniciaron las pruebas al paso o rápidas sobre la estructura base de la aplicación, construida a partir de UARGFlow, con un total de 47 pruebas iniciales. Posteriormente, se enfocaron los esfuerzos en los módulos correspondientes a la primera tanda de casos de uso. Aunque se priorizaron pruebas funcionales de tipo caja negra, también se realizaron pruebas de componentes, de valores extremos, unitarias, de usabilidad y de regresión, alcanzando un total de 45 pruebas en esta etapa, incluyendo pruebas

con fallo y de regresión. En la fase de construcción 2, correspondiente a la segunda tanda de casos de uso, se llevaron a cabo 50 pruebas, mientras que en la fase de construcción 3, asociada a la tercera tanda de casos de uso, se realizaron 16 pruebas. En ambos casos se incluyeron pruebas con fallo y de regresión.

- **Transición:** En esta etapa final se ejecutaron pruebas orientadas a garantizar la estabilidad y calidad del producto. Se realizaron pruebas finales, de integración y de regresión, totalizando 110 pruebas, de las cuales 18 presentaron fallos. Estas pruebas permitieron identificar y corregir errores críticos antes de la entrega final.

A continuación, se presenta el Gráfico 5, que resume la ejecución de los casos de prueba realizados a lo largo del desarrollo del sistema Testify.



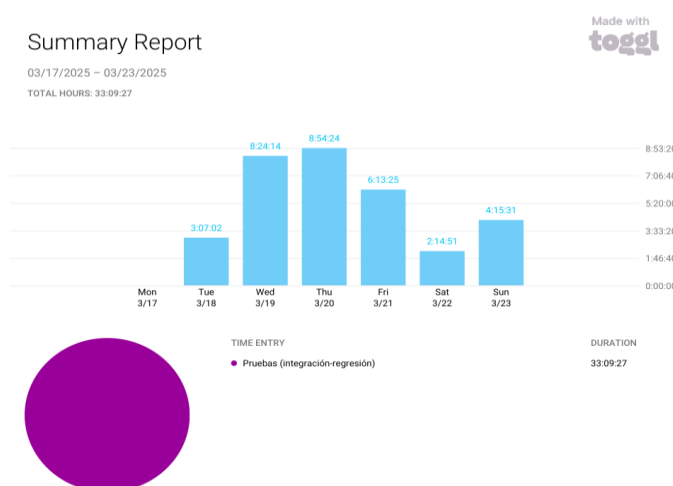
**Gráfico 5. Casos de prueba ejecutados**

El gráfico muestra la cantidad y diversidad de pruebas ejecutadas, lo que refleja el esfuerzo por alcanzar una amplia cobertura y asegurar el cumplimiento del plan de pruebas establecido. Esta ejecución sistemática permitió, además, mitigar riesgos

identificados previamente, contribuyendo al fortalecimiento de la calidad del producto final:

- RK86: No se detectan todos los errores a tiempo durante las pruebas del software.
- RK41: Las pruebas sobre los casos de uso llevan más tiempo del planificado.
- RK3: No se realizan suficientes pruebas al software.

Durante la etapa de transición se llevaron a cabo un total de 110 casos de prueba, entre los cuales se destacan las pruebas finales, de integración y de regresión. En conjunto, se destinaron 33 horas de trabajo, lo que representa un promedio estimado de entre 18 y 20 minutos por prueba. Este tiempo abarcó la preparación del entorno, la ejecución del caso, el análisis de resultados y la correspondiente documentación. De las 110 pruebas realizadas, 18 presentaron fallos, lo que permitió detectar y corregir errores críticos antes de la entrega final del sistema. Es relevante destacar estos resultados, ya que reflejan el compromiso del equipo con la calidad del proceso de validación: todas las pruebas debieron ejecutarse nuevamente en su totalidad, incluyendo además nuevos casos, con el objetivo de asegurar la estabilidad y confiabilidad del producto final. El gráfico extraído de TogglTrack (Gráfico 6) resume el tiempo dedicado a esta fase de pruebas.



**Gráfico 6. Tiempo dedicado a fase de pruebas.**

De este modo, se cumple con el objetivo principal del plan de pruebas: asegurar la ejecución efectiva de las pruebas necesarias, centradas en los componentes clave de la aplicación, a fin de garantizar su correcto funcionamiento conforme a lo previsto y en línea con los requerimientos del cliente.

### ***Gestión SQA para el aseguramiento de la calidad del software***

El Plan de Calidad implementado durante el desarrollo de la aplicación Testify fue diseñado para cumplir con los estándares, nomenclaturas y criterios definidos por el equipo de desarrollo OSLO, con el fin de garantizar la calidad del producto final.

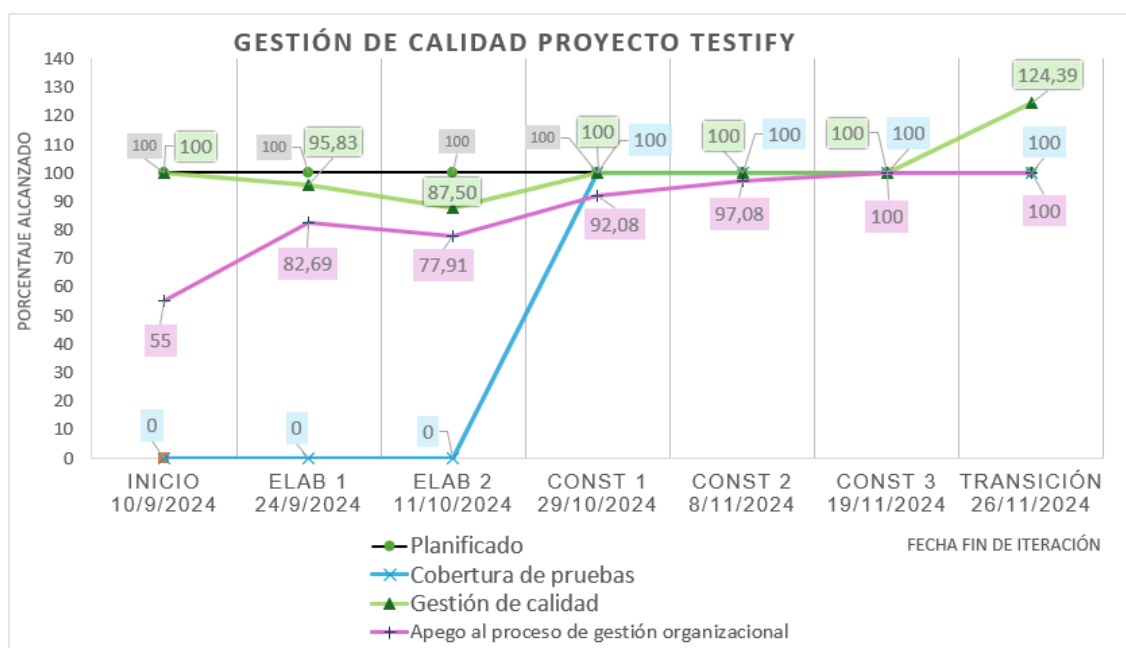
Para ello, se controló especialmente la confección de los documentos considerados “clave”, así como también de la documentación de apoyo. Si bien esta última no incide directamente en el funcionamiento del software, los documentos clave sí tienen un impacto directo sobre el producto. Por este motivo, se realizaron múltiples revisiones de documentos a lo largo de las distintas etapas del modelo RUP. A continuación, se describen dichas actividades por fase:

- **Inicio:** En esta etapa se priorizaron revisiones generales de documentos, basadas en criterios como ortografía, estándares definidos por el equipo OSLO (colores, márgenes, logos) y la nomenclatura de nombres establecida para los entregables y artefactos en el repositorio de GitHub. También se revisaron el formato y las convenciones utilizadas para realizar commits y push desde los repositorios locales al repositorio colaborativo, incluyendo el uso correcto del nombre de la rama (documentación, software o main), el tipo de cambio (Gestión SQA, Pruebas, Correcciones, Requerimientos) y una descripción detallada de las modificaciones. Este control riguroso tuvo como objetivo evitar conflictos entre ramas o facilitar su resolución en caso de que ocurrieran, gracias a commits bien estructurados.
- **Elaboración:** En esta etapa, y dada su naturaleza, se comenzaron a revisar documentos considerados clave para el desarrollo del sistema. Uno de los más

relevantes fue la Especificación de Requerimientos de Software, cuya evaluación se basó en múltiples criterios de contenido, tales como: ¿Existen descripciones poco claras o sin funcionalidad?, ¿Faltan requerimientos funcionales?, ¿Se cumple con un mínimo de trazabilidad?, ¿Los requerimientos fueron verificados con el cliente? ¿Existen requerimientos que tienen mayor prioridad que otros?. Además de estos criterios funcionales, también se consideraron aspectos formales como el cumplimiento del estándar OSLO, la ortografía y la presencia de un índice funcional claro y ordenado. Otro documento clave revisado fue el Modelo de Casos de Uso (MCU), para el cual se aplicaron criterios como: ¿Está anexado el modelo al documento entregable del MCU? ¿Existe ambigüedad entre los casos de uso? ¿Hay actores no identificados? ¿Qué grado de aceptación tiene el modelo por parte del cliente? También en este caso se aplicaron criterios formales similares: estándar OSLO, ortografía e índice funcional. Asimismo, a lo largo del proceso de desarrollo de Testify, se revisaron otros documentos clave, como el Modelo de Diseño, la Arquitectura del Software, el Plan de Riesgos y el Plan de Pruebas, junto con documentos de apoyo. Cada uno de estos fue evaluado en función de criterios específicos, definidos por el equipo y ajustados a las buenas prácticas propuestas por la metodología.

Este esquema de revisiones se sostuvo a lo largo de las tres fases de la etapa de Construcción y durante la fase de Transición, quedando plasmado en los documentos entregables correspondientes: Revisión de SQA, Informe Final de SQA, Revisiones Técnicas Formales y diversos artefactos asociados, organizados en planillas de Excel.

El gráfico 6 resume las distintas actividades de gestión de calidad y pruebas de software realizadas en cada etapa del proceso de desarrollo.



**Gráfico 6. Gestión de Calidad de Testify**

### ***Gestión de calidad, repositorio Github, y ajuste al proceso.***

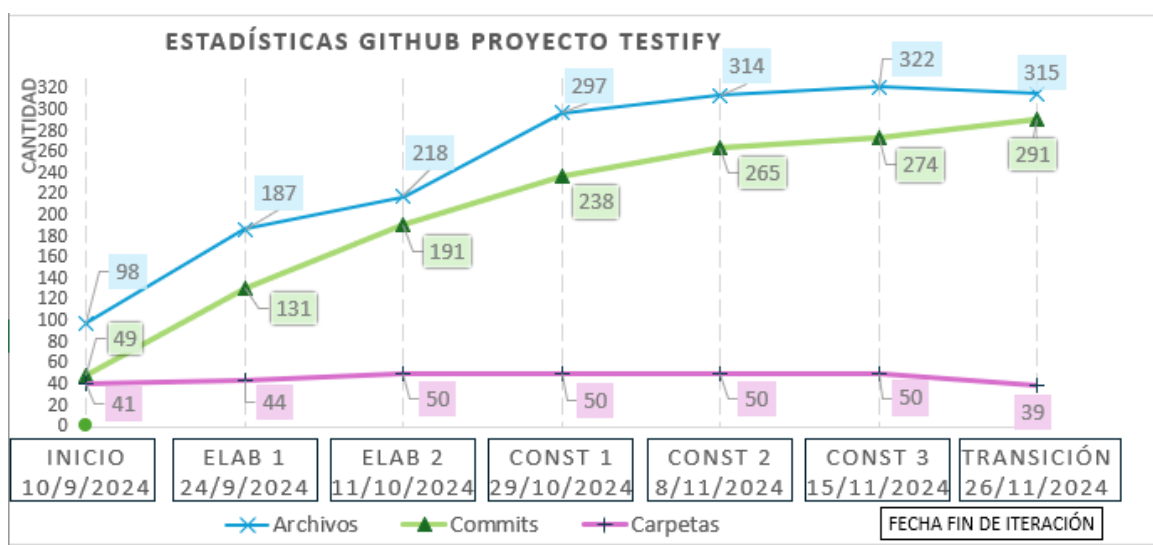
Desde el punto de vista de la gestión de calidad, en relación con el uso del repositorio GitHub y su integración con el proceso de desarrollo, se observó un crecimiento sostenido en la actividad de commits desde la etapa de Inicio hasta la Transición, reflejando un flujo de trabajo activo y constante (Gráfico 7).

Se destaca un incremento significativo entre las etapas de Inicio y Elaboración 2, donde los commits pasaron de 49 a 191. Este aumento se explica por la incorporación de una gran cantidad de documentación y entregables, el trabajo colaborativo entre los integrantes del equipo, las revisiones de SQA y las correspondientes correcciones. En contraste, durante las etapas de Construcción 1 y Construcción 3, el crecimiento de commits fue más moderado, lo que indica una mayor estabilidad en la implementación.

En cuanto al número de archivos, se observó un incremento progresivo desde la etapa de Inicio hasta Construcción 1. A partir de allí, el crecimiento se estabilizó,

alcanzando un total de 315 archivos al finalizar la etapa de Transición, tras una última revisión y limpieza del repositorio.

Respecto a las carpetas, el proyecto comenzó con 41 directorios y alcanzó un máximo de 50 durante la Elaboración 2, lo que refleja una consolidación en la estructura del repositorio. Finalmente, se cerró con 39 carpetas, producto de una reorganización y limpieza de directorios no utilizados durante la fase de Transición.



**Gráfico 7. Gestión de Calidad, GitHub y ajuste al proceso.**

Durante las etapas de Inicio y Elaboración 1, la relación entre cantidad de archivos y commits indica que se crearon o subieron muchos archivos con relativamente pocos commits. Esto se debió a que el equipo de desarrollo OSLO aún se encontraba en proceso de aprendizaje y adaptación al uso del repositorio GitHub. En esta etapa, los archivos eran subidos al repositorio solo una vez finalizados y revisados, sin respetar completamente el proceso propuesto de control de versiones, que implica realizar commits frecuentes en cada instancia de modificación (subir–hacer commit, modificar–hacer commit, revisar–hacer commit).

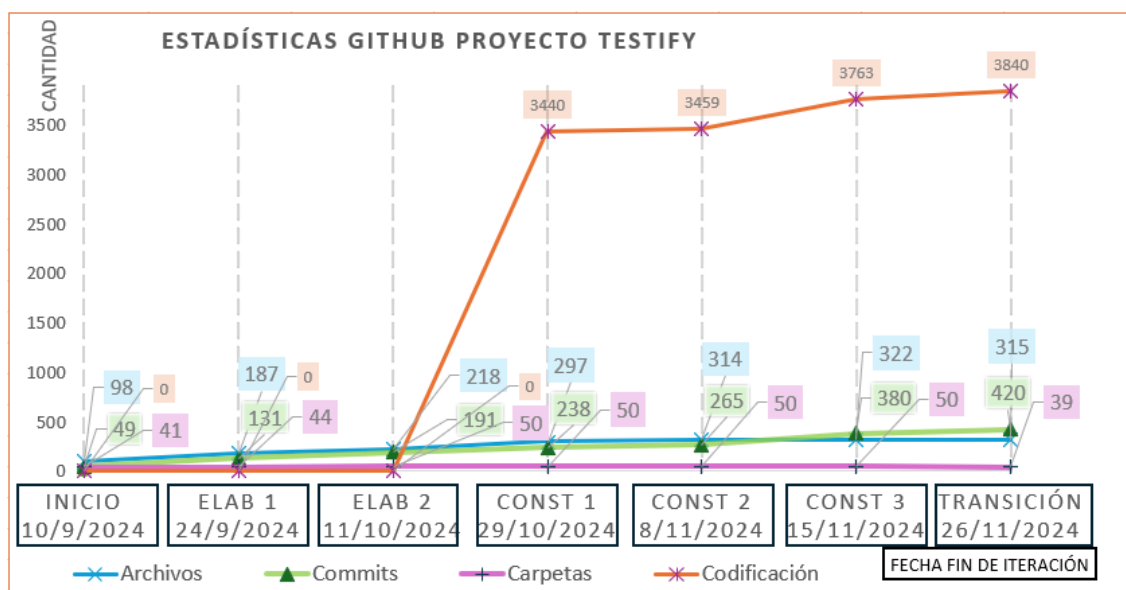
Esta práctica representaba un riesgo latente, ya que en caso de fallas técnicas (como equipos defectuosos o cortes de energía), el equipo podría haber perdido



documentación o entregables importantes que no habían sido respaldados en el repositorio.

A partir de la etapa de Elaboración 2 y hasta Construcción 3, se observó una mejora significativa en el proceso: la relación archivos/commits se estabilizó y se comenzó a registrar una mayor frecuencia de commits sobre archivos individuales. Esto refleja una correcta adopción de las prácticas de control de versiones y un mayor compromiso del equipo con el proceso establecido.

Por otra parte, la relación entre código y software mostró un desempeño constante durante las etapas de Construcción 1 y Construcción 2, lo cual indica un desarrollo sólido en torno a los requerimientos definidos. Asimismo, se evidenció un incremento sostenido en la cantidad de archivos hacia las etapas finales del proyecto, lo que da cuenta de una expansión progresiva y controlada del producto.



**Gráfico 8. Gestión de Calidad, GitHub y ajuste al proceso.**

## Tecnologías Utilizadas

Desde el inicio del proyecto, se definió una arquitectura tecnológica robusta y alineada con los objetivos funcionales y no funcionales de Testify. La solución fue concebida como un sistema web monolítico, compuesto por un frontend dinámico, un backend estructurado y una base de datos relacional.

Para el frontend se utilizó Angular 18, un framework basado en TypeScript que permite desarrollar aplicaciones de una sola página (SPA), con componentes reutilizables, alta interactividad y una integración fluida con servicios externos. Esta tecnología se complementó con Bootstrap, para la maquetación y los estilos, y con ng2-charts junto con Chart.js, para la visualización gráfica de datos, como los porcentajes de aprobación de escenarios.

En el backend, se optó por Java 17 junto con el framework Spring Boot 3.3.3, lo que permitió construir una API REST segura, escalable y de rápida implementación. El manejo de la persistencia se realizó mediante Hibernate, mientras que la seguridad fue abordada con Spring Security, lo que garantizó el control de accesos basado en roles y una estructura robusta para la autenticación.

El sistema de autenticación se integró con la API de Google, utilizando el protocolo OAuth2, lo que permitió a los usuarios autenticarse mediante sus cuentas institucionales de Gmail. Esta decisión no solo simplificó el acceso, sino que también incrementó la seguridad del sistema.

Como motor de base de datos se empleó MySQL 8.0, una opción eficiente y ampliamente adoptada en el desarrollo de aplicaciones web. El diseño de la base se realizó mediante un modelo entidad-relación, y se aplicó una normalización hasta la Tercera Forma Normal (3FN), a fin de evitar redundancias y garantizar la integridad de los datos.

Para la implementación, se utilizó Docker, herramienta que permitió empaquetar y ejecutar versiones UAT (User Acceptance Testing) del sistema, facilitando su entrega al cliente de forma controlada.

El entorno de desarrollo incluyó además herramientas como GitHub, para el control de versiones, y GitHub Desktop, para facilitar su uso por parte del equipo. Finalmente, para la distribución de software y documentación colaborativa, se empleó Google Drive, permitiendo un acceso ágil y ordenado a los materiales compartidos.

## Capítulo II: Fases del Proyecto

### Inicio

La Fase de Inicio comenzó en la segunda semana desde el inicio de la cursada de la asignatura Laboratorio de Desarrollo de Software. Durante esta etapa se abordaron temas cruciales para garantizar la continuidad y la finalización exitosa del proyecto.

En primer lugar, se definieron el nombre del equipo de trabajo y su logo identificador. A pesar de que los cuatro integrantes no nos conocíamos previamente, coincidimos de forma unánime con la propuesta realizada por uno de los miembros, lo cual evidenció desde el inicio una buena disposición al consenso. En este mismo momento, cada integrante asumió el compromiso formal de completar la cursada hasta el final del cuatrimestre, a pesar de contar con jornadas laborales de al menos siete horas diarias de lunes a viernes. Por ello, se discutió y acordó con claridad la responsabilidad que implicaba participar activamente en el proyecto.

Una vez establecidos los aspectos organizativos iniciales y teniendo definida la temática general de la propuesta a desarrollar durante 2024, comenzamos a reunir los datos necesarios para formalizar una entrevista con los clientes, a fin de avanzar en los requerimientos iniciales. A partir de la primera entrevista surgieron diversas dudas respecto al enfoque del proyecto, lo que motivó una segunda reunión. Esta instancia fue clave para delimitar el alcance del sistema y definir con mayor precisión el objetivo general.

Gracias a este proceso de validación, fue posible cumplir con los objetivos de la fase, elaborando los documentos correspondientes, tales como el Modelo de Negocio, el Estudio de Factibilidad, entre otros, que permitieron avanzar hacia la siguiente etapa del modelo RUP.

Esta fase estuvo marcada por el análisis, la adaptación y cierta incertidumbre inicial. Cada integrante debió encontrar su rol dentro del equipo y, en paralelo, apropiarse de

la propuesta a desarrollar. También fue necesario acordar la forma de trabajo a seguir, optimizando el uso del tiempo desde el principio. Afortunadamente, todos contábamos con alguna experiencia previa en trabajos grupales o en asignaturas similares, lo cual facilitó la integración y fortaleció la dinámica del equipo desde el comienzo.

## **Elaboración**

Durante la etapa de Elaboración del proyecto Testify, desarrollada entre el 9 de septiembre y el 7 de octubre de 2024, el equipo trabajó de manera progresiva y organizada a lo largo de dos iteraciones consecutivas.

### ***Inicio de la Elaboración – Iteración E01 (09/09 al 23/09)***

La Iteración E01 marcó el inicio efectivo de la fase de elaboración del proyecto Testify, y se extendió desde el 9 hasta el 23 de septiembre de 2024. En esta etapa nos enfocamos en reorganizar y replantear integralmente la propuesta de desarrollo inicial, a partir de las observaciones realizadas por la cátedra luego de la etapa de Inicio. Esto implicó redefinir el enfoque general del sistema, revisar los objetivos funcionales y técnicos, y ajustar el alcance del proyecto a condiciones más realistas en términos de tiempos, recursos y capacidades del equipo.

En la primera instancia de presentación, el 13 de septiembre, compartimos la nueva versión de la propuesta, en la cual organizamos los requerimientos funcionales en un total de 26 ítems (RF01 a RF26), haciendo hincapié en funcionalidades centrales como la creación de proyectos, la asignación de roles, el control de escenarios de prueba, la exportación de resultados, la visualización de métricas en tiempo real y la carga de documentos. Esta reformulación respondió a la necesidad de clarificar los procesos involucrados, simplificar circuitos internos y evitar redundancias. Al mismo tiempo, redefinimos los límites del proyecto y los roles del sistema, especialmente el del Gestor de Pruebas, que pasaría a tener un papel más activo en la creación y asignación de escenarios.

Durante esa misma semana elaboramos los primeros documentos de planificación: el Plan de Proyecto, el Plan de Calidad, el Plan de Gestión de Riesgos y el Plan de Configuración.

También se construyó una Matriz de Trazabilidad, herramienta esencial para el análisis y refinamiento de los requerimientos. Esta matriz tenía como objetivo asegurar que cada requerimiento funcional estuviera cubierto por al menos un caso de uso del sistema, y que no existieran elementos redundantes u obsoletos. En ella, cada fila representaba un caso de uso identificado por su nomenclatura específica, mientras que las columnas correspondían a los requerimientos funcionales. Las relaciones se marcaban mediante cruces en las intersecciones correspondientes. Aquellos requerimientos o casos de uso que no tuvieran vínculos eran objeto de análisis para su eliminación o reformulación. Este trabajo permitió depurar la especificación, mejorar la coherencia del diseño y garantizar una cobertura completa de las funcionalidades requeridas por el sistema. También seguimos utilizando con las herramientas colaborativas que nos acompañarían durante todo el desarrollo: Trello para gestión de tareas bajo metodología Kanban, y GitHub para el control de versiones del código fuente y los documentos del proyecto. Se definieron las tareas iniciales de la iteración y se comenzó a trabajar en la especificación técnica del sistema.

Para estimar el esfuerzo requerido, aplicamos el método de Puntos de Caso de Uso (PCU). En un primer cálculo, el resultado fue de 3040 horas-hombre, con una duración aproximada de 4,7 meses. No obstante, tras revisar en profundidad los requerimientos y redefinir la cantidad y complejidad de los casos de uso, esta estimación fue ajustada en sucesivas presentaciones. El 17 de septiembre presentamos una nueva versión del Plan de Estimación, que redujo el esfuerzo a 1216 horas-hombre, con una duración estimada de 1,9 meses. Este cambio fue posible luego de afinar el análisis de esfuerzo en cada etapa del desarrollo (análisis, diseño, programación, pruebas y sobrecarga) y de aplicar criterios más conservadores para evitar sobredimensionamientos.

En esa misma presentación, se expusieron las especificaciones técnicas del sistema: adoptamos una arquitectura monolítica con backend en Java + Spring Boot, frontend en Angular 18, base de datos MySQL gestionada con Hibernate, seguridad en Spring Security y autenticación con Google API mediante OAuth 2.0. También definimos otras herramientas complementarias como Bootstrap y Node.js para aspectos de diseño e instalación, respectivamente. Con esto establecimos una base tecnológica robusta y realista para el desarrollo del sistema.

Durante los días siguientes continuamos refinando las tareas de análisis, y para el 24 de septiembre, en la presentación de cierre de la iteración, presentamos una nueva reducción de esfuerzo: 950 horas-hombre y 1,5 meses de duración. A esta altura, también habíamos reducido la cantidad de casos de uso de 21 a 16, priorizando aquellos que eran esenciales para garantizar la funcionalidad mínima del sistema y dejar abierta la posibilidad de ampliación futura.

En esa presentación también se mostraron los avances en la Especificación de Requerimientos y en el Modelo de Casos de Uso. Ambos documentos fueron construidos de forma colaborativa y servirían como insumos para la siguiente iteración, en la que se profundizaría el diseño de base de datos, el modelo entidad-relación y el prototipo funcional.

Uno de los aspectos más importantes de la iteración fue el trabajo realizado en la gestión de riesgos. En total, se identificaron y trataron seis riesgos relevantes. Algunos de ellos fueron cerrados durante la iteración, como el RK04 (requerimientos mal definidos) y el RK60 (cambios en los objetivos del proyecto), gracias a entrevistas adicionales con el cliente y la entrega de documentación validada. Otros, como el RK3 (falta de pruebas suficientes), RK19 (cortes de electricidad o internet), RK20 (sobrecarga horaria) y RK40 (falta de capacitación técnica), se mantuvieron activos, pero fueron abordados con acciones de mitigación concretas. Estas incluyeron la definición del plan de pruebas, el resguardo diario de información, la organización semanal de tareas, la

distribución equitativa del trabajo, la comunicación anticipada de ausencias y la investigación autodidacta sobre tecnologías clave.

A nivel organizativo, mantuvimos un esquema de trabajo mixto, con reuniones virtuales por Google Meet y encuentros presenciales en el campus. Estas instancias nos permitieron tomar decisiones de forma consensuada, realizar ajustes sobre la marcha y mantener el compromiso del equipo.

También realizamos una revisión del repositorio GitHub, que funcionó como pilar del control de versiones del proyecto. En esta etapa inicial, el repositorio fue estructurado en ramas principales y secciones organizadas por entregables. Se comenzaron a subir los primeros archivos fuente y documentos, respetando la convención de nombres definida por el equipo OSLO. Además, se implementaron buenas prácticas de trabajo colaborativo, como el uso de commits descriptivos, actualizaciones frecuentes y sincronización entre los integrantes. Si bien aún no se cargaron componentes de código funcional del sistema, el repositorio ya contenía la documentación más importante generada durante la iteración, lo cual permitió mantener un registro ordenado y trazable desde el inicio del desarrollo.

Además, como parte del trabajo de planificación, pasamos de utilizar una única estimación global del esfuerzo a un enfoque más detallado basado en tres escenarios: optimista, más probable y pesimista. Esta estrategia nos permitió evaluar con mayor precisión los márgenes de error posibles, considerar distintos niveles de dificultad en la implementación y mejorar la toma de decisiones durante la planificación. Las tres estimaciones se construyeron a partir de los Puntos de Caso de Uso (PCU), considerando factores como el esfuerzo por disciplina (análisis, diseño, implementación y pruebas), la complejidad de cada caso de uso y los riesgos detectados. Esta perspectiva permitió proyectar un rango de esfuerzo más realista, con un mínimo estimado de 776 h-h, un valor más probable de 950 h-h y un máximo de 2455 h-h en el caso más desfavorable. Esta variabilidad fue clave para anticipar posibles desvíos y preparar escenarios de



contingencia. El uso de Trello permitió mantener visibilidad sobre las tareas asignadas y agilizó tanto la distribución como el seguimiento del trabajo.

Otro punto importante de esta iteración fue el tratamiento de los riesgos identificados. Se elaboró un plan inicial de gestión, se clasificaron los riesgos por nivel de severidad y se trazaron estrategias de mitigación. Entre los más relevantes estuvieron:

- RK3 (pruebas insuficientes): mitigado mediante la planificación del plan de pruebas.
- RK4 (requerimientos poco claros): resuelto tras entrevistas adicionales con el cliente.
- RK19 (problemas técnicos): enfrentado con backup diario y control de versiones actualizado.
- RK20 (sobrecarga horaria): se equilibraron las tareas en función de la disponibilidad del equipo.
- RK40 (falta de capacitación): cada miembro asumió la búsqueda de recursos y tutoriales sobre tecnologías claves.
- RK60 (cambios de alcance): cerrado luego de consensuar el objetivo final del sistema con el equipo docente.

Finalizamos esta etapa con una planificación sólida para la siguiente fase, una base técnica definida, un alcance más claro, una estimación de esfuerzo realista y una primera revisión del repositorio en GitHub. Todo esto nos permitió encarar la Iteración E02 con mayor seguridad, sabiendo que los fundamentos del sistema habían sido revisados y validados de forma rigurosa.

### ***Inicio de la Elaboración – Iteración E02 (23/09 al 07/10)***

La Iteración E02 del proyecto Testify se desarrolló entre el 23 de septiembre y el 7 de octubre de 2024, y tuvo como eje central el desarrollo técnico del sistema, el diseño

de su arquitectura funcional, la planificación de pruebas y la validación de los modelos construidos. Durante esta fase, el equipo avanzó de forma sostenida en la consolidación del modelo de datos, el prototipo funcional, el plan de pruebas y los entregables necesarios para cerrar la fase de Elaboración.

Durante los primeros días de trabajo de la iteración, se actualizó el Plan de Iteración E02, incluyendo la reorganización de tareas y su asignación de responsables. También se revisó el tablero de Trello y se evaluó el estado de avance de cada actividad. Uno de los principales focos fue la actualización del Plan de Contingencia, donde se reforzaron las estrategias de mitigación para riesgos activos como:

- RK3: realización insuficiente de pruebas.
- RK19: cortes de electricidad o internet.
- RK20: sobrecarga horaria.
- RK40: falta de capacitación técnica.

En este marco, se reorganizaron tareas, se propuso la redistribución de responsabilidades ante ausencias y se acordó la capacitación puntual en herramientas específicas para mejorar la eficiencia del equipo.

En esta etapa se detallaron las nuevas estimaciones de esfuerzo, reforzando el uso de tres escenarios: optimista, más probable y pesimista. Las cifras actualizadas oscilaron entre 606 h-h y 1918 h-h según el caso, lo cual reafirmó la validez del modelo de estimación adoptado. Además, se avanzó con el diseño del modelo entidad-relación de la base de datos y se revisó el prototipo funcional.

A lo largo de los días siguientes, se consolidó el modelo de datos definitivo y se mejoró el prototipo funcional realizado en Excel. Este prototipo incorporó cambios y permitió validar la coherencia entre los flujos de navegación, la interfaz de usuario y los requerimientos funcionales definidos.

Al cierre de la iteración, se completó el Plan de Pruebas, que incluyó el detalle de casos de prueba por característica y por caso de uso. Se establecieron prioridades, criterios de aprobación, entornos de prueba, tipos de prueba (funcionales, regresión, rendimiento) y clasificación de errores. También se identificaron riesgos adicionales asociados al tiempo de pruebas y a la capacitación del personal, proponiendo medidas de contingencia adecuadas.

En paralelo, se elaboró el Modelo de Diseño, estructurando el sistema en subsistemas como Usuarios y Roles, Casos de Prueba, Proyectos, Adjuntos, Iteraciones y Exportación. Este modelo sentó las bases arquitectónicas para la siguiente fase. Además, se realizaron Revisiones Técnicas Formales (RTF) y actividades de SQA, en las cuales se detectaron observaciones menores que fueron corregidas, fortaleciendo la trazabilidad y la calidad de los documentos entregables.

La Iteración E02 concluyó con la validación de los principales artefactos de diseño y prueba, una arquitectura clara, y una planificación sólida de cara a la fase de Construcción. Se cumplió con todos los objetivos previstos para la iteración, incluyendo la finalización del modelo de datos, el plan de pruebas, el modelo de diseño y la gestión de riesgos. Se incorporaron nuevos casos de uso como CRUD Tipo, CRUD Subtipo y CRUD Categoría, llevando el total a 18 casos definidos. Las actividades planificadas se ejecutaron dentro de los plazos establecidos, incluyendo las revisiones de SQA y el cierre formal de la iteración. También se verificó la evolución del repositorio: se pasó de 181 documentos y 52 carpetas al inicio, a un total de 187 documentos y 294 carpetas al cierre de la iteración. Estos resultados reflejan una evolución constante en la producción y organización de los entregables del proyecto, lo que refuerza la calidad del proceso y nos deja en condiciones de avanzar sin necesidad de replanteos mayores. La base establecida nos permitió avanzar hacia la implementación con confianza en la coherencia técnica y organizativa del sistema. La fase de elaboración nos dejó preparados para encarar la construcción con un sistema bien definido, diseñado y con sus principales riesgos bajo control.

## Construcción

### *Fase de Construcción 1*

La primera fase de construcción, comprendida entre el 15/10/2024 y el 29/10/2024, marcó el inicio del desarrollo de los primeros módulos de Testify, continuando con la estructura previamente esbozada en el proyecto UARGflow.

En cuanto al uso o adaptación del sistema UARGFlow, es importante señalar que esta posibilidad fue considerada en las primeras etapas del proyecto, específicamente durante la Fase de Inicio, al momento de explorar alternativas tecnológicas y arquitectónicas que pudieran ser reutilizadas o adaptadas para la construcción de Testify. Sin embargo, luego de una evaluación preliminar por parte del equipo OSLO, se concluyó que la incorporación de UARGFlow al proyecto implicaba una inversión de tiempo considerable, tanto para su estudio en profundidad como para su posterior adaptación a los requerimientos específicos del sistema de gestión de pruebas que estábamos desarrollando.

En concreto, se identificaron varios factores que hacían inviable su adopción en el contexto de trabajo propuesto. Por un lado, UARGFlow no contaba con una documentación suficientemente detallada ni con una estructura modular que facilitara su integración parcial o la reutilización directa de sus componentes. Esto implicaba que el equipo debía destinar una cantidad significativa de horas a la comprensión del funcionamiento interno del sistema, incluyendo su lógica de negocio, estructura de datos y dependencias técnicas.

Por otro lado, una vez comprendido su funcionamiento, debía llevarse a cabo un proceso de análisis, selección y adecuación de los componentes reutilizables, lo que requería una etapa de adaptación no menor, dado que la lógica propia de un gestor de trámites (funcionalidad original de UARGFlow) no se alineaba directamente con los objetivos ni con el flujo de trabajo esperado para un sistema orientado a la gestión de casos de prueba.

Al comparar estos requerimientos con el cronograma general del proyecto, se observó que los tiempos que demandaría esta tarea excedían ampliamente los plazos disponibles para el desarrollo de Testify. Dado que el proyecto debía completarse dentro de un cuatrimestre, y considerando las múltiples fases del modelo RUP, se priorizó avanzar con una arquitectura construida desde cero, más acotada y adaptada específicamente a las funcionalidades necesarias, garantizando así una mayor eficiencia en la implementación y una mejor distribución de los esfuerzos del equipo.

Si bien la reutilización de UARGFlow fue contemplada inicialmente como una opción, se tomó la decisión fundamentada de no utilizarla debido a la desproporción entre el esfuerzo que implicaba su adaptación y los beneficios que podría aportar, en relación con los recursos y tiempos disponibles para el desarrollo de Testify.

Gracias a la experiencia previa de uno de los integrantes en desarrollo de software, las tareas de documentación, análisis y pruebas fueron distribuidas entre el resto del equipo, permitiendo una organización más eficiente del trabajo.

Durante esta etapa, se llevó a cabo una capacitación intensiva en el uso de diversas herramientas, tales como GitHub, Trello, Canva y Visual Paradigm, así como en tecnologías como Angular y MySQL. Esta formación resultó fundamental para que el equipo alcanzara con éxito los objetivos planteados para esta iteración.

En relación con la documentación requerida, se cumplió en tiempo y forma con los entregables solicitados, y en algunos casos fue posible adelantarlos. Esta planificación anticipada permitió al equipo enfocarse con mayor profundidad en las tareas de desarrollo, logrando un equilibrio adecuado entre la producción de documentación y el avance técnico del proyecto.

Se han realizado un total de 45 pruebas sobre los módulos Categoría, Tipo, Subtipos e Iteraciones. Se han detectado 14 errores durante la etapa de pruebas, los cuales fueron corregidos el 100% de ellas durante la misma fase.

### ***Fase Construcción 2***

La segunda fase de construcción se desarrolló entre el 29/10/2024 y el 08/11/2024. Durante esta iteración, se avanzó con el desarrollo de los módulos previstos, cumpliendo con los plazos establecidos tanto en lo técnico como en lo documental.

Al inicio de esta fase, uno de los integrantes debió ausentarse temporalmente por motivos laborales, lo cual le impidió participar de las presentaciones en clase. Ante esta situación, el equipo decidió redistribuir las tareas, asumiendo solidariamente las responsabilidades del integrante ausente.

Las actividades de desarrollo y pruebas se llevaron adelante con normalidad y dentro de los plazos acordados. En esta etapa, se evidenció un mayor nivel de cohesión en el equipo, lo que favoreció la ejecución fluida de las tareas y una mejor adaptación frente a eventuales cambios, optimizando así el rendimiento colectivo.

Se han realizado un total de 50 pruebas sobre el módulo Escenarios. Se han detectado 25 errores durante la etapa de pruebas, los cuales fueron corregidos el 80% de ellas durante la misma fase, y el 20% restante fué corregido durante la siguiente etapa de construcción.

### ***Fase Construcción 3***

Esta iteración, llevada a cabo entre el 08/11/2024 y el 19/11/2024, presentó diversas dificultades, principalmente en lo relativo a la comunicación interna del equipo. Uno de los integrantes tuvo que viajar nuevamente, lo que limitó su disponibilidad horaria y generó complicaciones derivadas de la diferencia de huso horario.

Estas condiciones afectaron directamente la coordinación de tareas, lo cual impactó en la organización de las presentaciones correspondientes a la iteración. El desarrollo de los módulos previstos se concretó hacia el final de la fase y se extendió incluso al inicio de la siguiente etapa (Transición).

No obstante, frente a estos desafíos, el equipo optó por aprovechar el tiempo disponible para avanzar en los planes de prueba que se ejecutarían una vez concluido el desarrollo. Asimismo, los integrantes responsables de la documentación comenzaron a trabajar en el diseño de los documentos que el sistema debía generar, anticipando tareas y optimizando el uso del tiempo disponible.

Se han realizado un total de 16 pruebas sobre el módulo Exportación de Proyectos y Consultar Proyectos asignados. Se han detectado 4 errores durante la etapa de pruebas, los cuales fueron corregidos el 30% de ellas durante la misma fase, y el 70% restante fué corregido durante la siguiente etapa de Transición.

Durante el transcurso de esta iteración, se procedió con el comienzo de los manuales de Instalación y manual de usuario.

### ***Conclusión Fase Construcción***

El trabajo realizado por el equipo OSLO durante las tres fases de construcción no sólo permitió avanzar significativamente en el desarrollo del sistema, sino que también representó una experiencia enriquecedora en la aplicación de metodologías ágiles, gestión de proyectos, gestión de riesgos y aprendizaje de nuevas tecnologías.

La disciplina, el compromiso y la capacidad de adaptación del equipo fueron factores clave para el éxito de esta etapa. El cumplimiento de los objetivos establecidos y la alineación del trabajo con la planificación inicial demuestran la importancia de una ejecución organizada, colaborativa y orientada a resultados.

## Transición

La fase de transición se desarrolló a partir de la conclusión de la tercera etapa de construcción y de la asignatura laboratorio de desarrollo de software, si bien se iniciaron borradores del manual de instalación y el manual de usuario durante la cursada para obtener retroalimentación su finalización total ocurrió luego del cuatrimestre, convirtiéndose en un aspecto clave de esta etapa, sumado a ello el objetivo principal fue llevar el sistema desde un entorno de desarrollo hasta su puesta en funcionamiento/producción mediante el despliegue y configuración en un entorno simulado para los usuarios finales. Durante esta fase además se realizaron pruebas finales de integración, regresión y validación con su respectiva carga de datos y pruebas destinadas a la detección y corrección de errores pasando por múltiples versiones de la aplicación Testify hasta alcanzar una estabilización constante del sistema.

Este conjunto de actividades llevadas a cabo durante la fase de transición apoyado por una excelente comunicación y trabajo colaborativo entre los integrantes del equipo de desarrollo OSLO permitió finalizar con el objetivo planteado al inicio de la cursada: el desarrollo de una aplicación de gestión y control de pruebas de software funcional, es por ello que para facilitar su implementación se optó por el uso de Docker mediante imágenes y contenedores lo que garantizará una entrega eficiente y eficaz en respuesta a los desafíos del despliegue y compatibilidad.

En relación con las estimaciones realizadas, puede afirmarse que los tiempos reales de trabajo fueron ajustándose a lo planificado en cada iteración, con el propósito de asegurar el cumplimiento de los plazos establecidos. El equipo asumió un alto grado de compromiso y dedicación, lo que permitió que el desarrollo efectivo del proyecto se mantuviera en una línea de consistencia respecto de las estimaciones iniciales. Esta correspondencia entre lo planificado y lo ejecutado refleja una adecuada capacidad de adaptación y gestión del tiempo por parte del equipo de trabajo.

Tras la recepción formal de las observaciones por parte del equipo docente el día 31 de marzo, el equipo OSLO se abocó durante el mes de junio a la revisión, ajuste y



mejora integral de la documentación, el código fuente y los entregables asociados al sistema Testify.

Entre las tareas más relevantes realizadas se encuentran:

- Actualización del Manual de Usuario: Se corrigieron errores ortográficos y gramaticales, se mejoró la redacción general, se actualizaron las capturas de pantalla y se reestructuraron secciones clave. Además, se incorporó una explicación más clara sobre los objetivos del sistema, los roles disponibles, y el flujo de trabajo recomendado para su uso, basado en un proyecto completo de prueba.
- Revisión del Manual de Instalación: Se completaron datos faltantes y corrigieron cuestiones de forma.
- Carga de datos reales para pruebas: Se cargaron proyectos completos en la base de datos para facilitar la validación funcional del sistema. Se resolvieron problemas con caracteres acentuados y se probaron los informes exportados incluyendo distintos escenarios, iteraciones y estados.
- Limpieza y actualización del repositorio GitHub: Se eliminaron documentos en estado borrador, se reorganizaron los archivos README. Se completó la documentación pendiente de la etapa de transición y se agregaron gráficas de actividad semanal, esfuerzo aplicado y estadísticas de Git.
- Memoria del Proyecto: Se amplió el análisis retrospectivo de los desafíos enfrentados por el equipo, incluyendo las dificultades para definir requisitos, justificar la no implementación de UARGFlow, y se detallaron aportes de gestión como la categorización de pruebas, mejoras en la experiencia del Tester y ajustes realizados al proceso de commits en GitHub. También se incluyó un análisis sobre los riesgos detectados y activados, junto con sus respectivas mitigaciones.

Estas acciones no sólo permitieron responder adecuadamente a todas las observaciones recibidas, sino que también representaron una oportunidad valiosa para

consolidar aprendizajes, fortalecer la documentación del proyecto y aumentar la calidad general del producto entregado.

## Conclusiones

### Conclusión individual Malena Oyarzo

Luego de haber cursado esta asignatura en el 2024 y estar en esta instancia del tramo final, creo que valió la pena el esfuerzo realizado. Al inicio de la cursada no fue fácil formar parte de algún equipo, porque cuando no estás en un grupo de estudio, la consigna es adaptarte al grupo de trabajo o estudio que te toque y en esta asignatura uno de los objetivos es trabajar en equipo para desarrollar un producto de software. Y si bien al principio no fue fácil encastrar con mis compañeros del equipo OSLO, a medida que avanzó la cursada cada uno fue encontrando su lugar y de esa forma con el esfuerzo de cada uno fuimos logrando objetivo tras objetivo, el compromiso fue claro desde el principio y eso se notó como equipo. Más allá de nuestros trabajos y ocupaciones personales siempre buscamos la forma de llegar a tiempo con los objetivos en cada clase, lo que nos fortaleció para seguir avanzando hasta el final de la cursada.

En lo personal fue una experiencia satisfactoria, la propuesta que tuvimos que desarrollar con los otros dos equipos me pareció súper interesante, porque además sumó para entender con más claridad procesos del desarrollo de un producto de software que son sumamente importante realizarlos porque fortalece la calidad del trabajo que se lleva adelante, te permite prever aspectos que si te dedicas solo al análisis, diseño, y programación no los ves, causando inconvenientes innecesarios que te atrasan en el desarrollo del producto.

En esta cátedra se utilizan todas las herramientas adquiridas en las asignaturas de los años anteriores y si bien la cursada dura solo 15 semanas aproximadamente, en ese corto tiempo nos organizamos con la guía del equipo docente, para fortalecer e implementar toda la teoría aprendida anteriormente. En lo particular, el tema de las

presentaciones casi todas las clases, me parece algo muy productivo porque ayuda al alumno, tanto al que tiene ya experiencia como al que no, a desenvolverse, a perder la timidez y a adquirir el vocabulario adecuado al momento de presentar un trabajo.

Para finalizar, solo me queda decir que fue una experiencia muy grata esta cursada con mi equipo de trabajo OSLO, tuvimos buenos y malos momentos, pero siempre encontramos la solución para salir adelante juntos, también resultó muy agradable compartir con los otros dos equipos de trabajo, porque pese a las diferentes edades que tenemos logramos una armonía de trabajo agradable donde todos aprendimos de todos.

### **Conclusión individual Emilio Levipichun**

Durante el año 2018 tras no poder finalizar la materia laboratorio de desarrollo de software sin compañeros de equipo, logro destacar que poder finalizar esta vez en el año 2024-2025 con un equipo que mantuvo siempre la comunicación y coordinación de forma constante hizo que fuera una experiencia enriquecedora, a pesar de no conocernos previamente logramos formar vínculos y trabajar de forma colaborativa.

Uno de los aspectos que más destaco (y que considero una de las mayores fortalezas como grupo) es que todas las decisiones fueron tomadas a nivel grupal y no unilateral como me ocurrió en materias anteriores donde la falta de compromiso o interés afectaban al resultado final, además esta materia me enseñó a desenvolverse con seguridad en presentaciones orales, entendiendo que además del software o de los documentos entregables una buena presentación es una actividad que requiere preparación, investigación, escritura, ejemplos y no algo que se debía improvisar cinco minutos antes de exponer, así como también me enseñó a valorar, confiar y colaborar de forma más activa con mis compañeros contribuyendo de forma eficiente a un objetivo en común.

### **Conclusión Individual Valeria Ojeda Muñoz**

Desde mi experiencia como parte del equipo de desarrollo de Testify, puedo decir que esta asignatura representó el cierre de un ciclo importante en mi formación. La experiencia de trabajar en equipo fue muy enriquecedora y superadora en comparación con otras. A diferencia de proyectos anteriores, sentí que hubo más comunicación entre los miembros del equipo, una mejor distribución de roles y un compromiso real de cada integrante. Cada uno enfrentó sus propios desafíos; en mi caso, el principal fue lograr organizarme entre mis responsabilidades laborales y el desarrollo del proyecto. Sin embargo, al ser honestos entre nosotros y compartir nuestras situaciones, pudimos coordinar y cumplir con las iteraciones sin mayores demoras.

El uso de herramientas colaborativas como Trello y GitHub fue clave para sostener ese trabajo ordenado, comunicativo y eficiente. Además, todos aprendimos cosas nuevas, ya sea en lo técnico o en lo organizativo. Personalmente, me animé a proponer la utilización de la matriz de trazabilidad, una herramienta que terminó siendo muy útil para darnos cuenta de que era necesario hacer modificaciones al proyecto. Ver que esa propuesta tuvo impacto en el proceso me dejó una sensación de orgullo y de crecimiento personal. Estoy muy contenta por lo que logramos como grupo y por haber cerrado esta etapa académica con una experiencia de trabajo tan positiva y formativa.

### **Conclusión individual Eduardo Sly**

Como todo proyecto en un ámbito laboral, y más cuando uno cambia de empresa, uno comienza el mismo sin conocer en totalidad al resto de los compañeros de trabajo. Lo mismo aplicó durante el comienzo del equipo OSLO. Sin embargo, formar parte del equipo OSLO resultó una experiencia muy enriquecedora desde donde se aprendió que cuando hay personas que asumen su rol desde el día uno y contribuyen con su conocimiento, no importa el seniority del integrante para lograr sacar un proyecto adelante. Con el paso de los días, el equipo logró consolidarse como un grupo de trabajo sólido, combinando tanto la experiencia de mis compañeros en proyectos previos de la materia como la que pude aportar desde mi ámbito laboral, lo que nos llevó a tener una buena comunicación, distribución de tareas y colaboración retroactiva entre todos los integrantes.

Sobre la materia, el contenido me pareció muy valioso, ya que invita a los alumnos a tener su primera experiencia en el desarrollo de un proyecto de desarrollo de software y a cómo generar el compromiso de enfrentar un desafío como un emprendimiento.

## Conclusión Final

La experiencia de desarrollar Testify representó mucho más que la simple concreción de un producto de software. Fue un proceso formativo, humano y profesional que dejó una experiencia significativa en cada integrante del equipo OSLO. A lo largo de la cursada, cada estudiante enfrentó desafíos propios, pero también supo reconocer el valor del trabajo colaborativo como herramienta para sortear dificultades y alcanzar objetivos en común.

El inicio del proyecto no estuvo exento de obstáculos: integrarse a un nuevo equipo, asumir roles y construir dinámicas de trabajo no siempre es sencillo. Sin embargo, con el correr de las semanas, OSLO se consolidó como un grupo comprometido, con buena comunicación, distribución equilibrada de tareas y una gran capacidad para resolver problemas en conjunto.

En términos de aprendizaje, la cursada permitió articular de forma concreta los contenidos teóricos adquiridos en años anteriores, aplicándolos en un entorno realista de desarrollo. Además, las instancias de presentación y exposición fomentaron habilidades comunicativas clave, brindando seguridad, vocabulario técnico y confianza al momento de compartir avances y decisiones del proyecto.

En relación con la dificultad para definir los requisitos y presentar alternativas de diseño en las primeras etapas del desarrollo, reconocemos que fue uno de los principales desafíos que enfrentó el equipo durante la Fase de Inicio y parte de la Fase de Elaboración del proyecto Testify. Esta situación fue abordada activamente mediante

un proceso iterativo de diálogo y revisión conjunta con los clientes, en el que ambas partes trabajaron en clarificar las necesidades funcionales y operativas del sistema.

Es importante destacar que, al tratarse de un sistema nuevo que venía a cubrir una necesidad no resuelta hasta el momento en la materia, los propios requerimientos no se encontraban totalmente definidos ni sistematizados al inicio del proyecto. Esto es comprensible, ya que en muchos casos las necesidades se vuelven más visibles y precisas en el proceso de modelado y validación, a medida que se comienzan a construir representaciones tangibles del sistema. En este sentido, desde el equipo OSLO se promovió un enfoque de trabajo colaborativo, proponiendo diversas instancias de validación mediante entrevistas, reuniones presenciales y la elaboración de prototipos de diseño.

Durante las primeras semanas, se realizaron dos entrevistas clave con los clientes, con el objetivo de relevar en profundidad las funcionalidades esperadas y detectar tanto requerimientos explícitos como necesidades implícitas. Estas instancias permitieron establecer una base de entendimiento compartido, aunque en algunos casos surgieron diferencias en la interpretación de ciertos conceptos o prioridades, lo que llevó a continuar afinando los requisitos en etapas posteriores.

En cuanto al diseño de la interfaz y de la experiencia de usuario, se presentaron múltiples alternativas mediante prototipos desechables desarrollados en Figma, y en algunos casos mediante maquetas rápidas en Excel. Si bien algunas de estas propuestas no recibieron una validación formal inmediata por parte de los clientes, resultaron fundamentales como punto de partida para el análisis conjunto, contribuyendo a ir ajustando paulatinamente tanto el enfoque funcional como estético del sistema.

En retrospectiva, consideramos que este proceso de definición y ajuste de los requerimientos fue enriquecedor, ya que permitió mejorar la comunicación con los clientes, fortalecer las competencias del equipo en la gestión de la incertidumbre y,

sobre todo, garantizar que las funcionalidades implementadas en Testify respondieran a necesidades reales del entorno académico.

En relación con los aportes realizados por el equipo OSLO al diseño funcional y a la gestión del sistema Testify, consideramos importante destacar dos decisiones clave que, si bien pueden parecer sutiles en el producto final, representaron aportes significativos durante el proceso de desarrollo, especialmente en lo que respecta a la trazabilidad, la usabilidad y la experiencia del usuario final.

Por un lado, se incorporó la posibilidad de categorizar los casos de prueba, es decir crear un tipo y subtipo asociado, funcionalidad no contemplada en la propuesta inicial. Esta característica permitió organizar y clasificar los casos de prueba de forma más estructurada, facilitando el seguimiento, análisis y recuperación de información durante las distintas fases del ciclo de vida del software. La inclusión de esta funcionalidad respondió a la necesidad de mejorar la trazabilidad de las pruebas ejecutadas, permitiendo al equipo identificar de manera rápida qué pruebas correspondían a determinado tipo, subtipo o categoría funcional dentro del sistema. Este aporte surgió a partir del análisis conjunto de herramientas de benchmarking, como TestRail y PractiTest, y se consolidó como una solución propia para mejorar la visibilidad y gestión de los escenarios en Testify.

Por otro lado, se realizaron adaptaciones específicas en la interfaz destinada al rol de Tester, diferenciando sus vistas y funciones de aquellas asignadas a otros perfiles, como el Gestor de Pruebas. Esta decisión buscó ofrecer al usuario Tester una experiencia más clara, directa y eficiente, acotando la cantidad de opciones visibles y priorizando las funcionalidades asociadas a sus tareas principales: la ejecución de pruebas, la carga de resultados y la inclusión de observaciones. Esta personalización de las pantallas implicó repensar el diseño inicial del sistema para adaptarlo a los flujos reales de trabajo, lo cual representa un aporte en términos de diseño centrado en el usuario, contribuyendo a una mayor usabilidad y reducción de errores operativos.

Ambas decisiones fueron tomadas a partir de discusiones técnicas internas y en función de observaciones recibidas durante la cursada, y constituyen aportes funcionales concretos que mejoraron la calidad del sistema Testify, tanto desde la perspectiva de gestión como desde la experiencia del usuario.

Un obstáculo técnico relevante surgió durante la integración del framework Bootstrap en el frontend de la aplicación. Inicialmente, se intentó utilizar la última versión disponible de Bootstrap, con el fin de aprovechar sus mejoras visuales y componentes actualizados. Sin embargo, se identificó que dicha versión no era plenamente compatible con Angular 18, versión con la que se estaba desarrollando Testify, debido a discrepancias en la gestión de dependencias y estilos. Como resultado, el equipo optó por utilizar una versión anterior y estable de Bootstrap, que sí ofrecía compatibilidad, y se realizaron los ajustes manuales necesarios en los estilos y componentes para mantener una experiencia de usuario fluida sin comprometer la funcionalidad.

Más allá de lo técnico, cada testimonio deja ver un profundo crecimiento personal. Testify fue el resultado de muchas manos, ideas y esfuerzos, pero también la muestra de que el respeto, la escucha y la colaboración genuina son pilares fundamentales para cualquier trabajo grupal exitoso.

En definitiva, esta asignatura no solo cerró un ciclo académico, sino que dejó enseñanzas valiosas que cada integrante del equipo OSLO llevará consigo, tanto en lo profesional como en lo humano.