



UNIVERSIDAD NACIONAL
de MAR DEL PLATA



Facultad de Ingeniería
Universidad Nacional de Mar del Plata

***Implementación del sistema GRIN Global
Community Edition para la gestión de los
bancos de
germoplasmas de la Red de Recursos
Fitogenéticos del INTA***

Alumno: Escalante, Santiago

Director: Mg. Lic. Genin, Fernando

Co-Director: Ing. Spinelli, Adolfo Tomás

Proyecto Final para optar al grado de Ingeniero Informático de la
Facultad de Ingeniería de la Universidad Nacional de Mar del Plata

Mar del Plata, 30 de octubre del 2024



RINFI es desarrollado por la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución- NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).



UNIVERSIDAD NACIONAL
de MAR DEL PLATA



Facultad de Ingeniería
Universidad Nacional de Mar del Plata

***Implementación del sistema GRIN Global
Community Edition para la gestión de los
bancos de
germoplasmas de la Red de Recursos
Fitogenéticos del INTA***

Alumno: Escalante, Santiago

Director: Mg. Lic. Genin, Fernando

Co-Director: Ing. Spinelli, Adolfo Tomás

Proyecto Final para optar al grado de Ingeniero Informático de la
Facultad de Ingeniería de la Universidad Nacional de Mar del Plata

Mar del Plata, 30 de octubre del 2024

ÍNDICE

Agradecimientos	4
Resumen	5
Introducción	7
Objetivos del proyecto	8
Alcance del proyecto	8
Soporte	9
Impacto	9
Entregables	10
Análisis del problema	11
Dominio del problema	11
Actores del sistema	12
Problema a Resolver	13
Aporte del proyecto	14
Parte interesada	14
El equipo	14
Análisis FODA	15
Fortalezas	15
Oportunidades	15
Debilidades	16
Amenazas	16
Análisis de Riesgos	17
Medidas de contingencia	19
Estimación inicial	20
Metodologías	23
Metodología de trabajo	23
Metodología de implementación	24
Metodología de pruebas	25
Diseño del sistema	27
Arquitectura	27
Arquitectura simplificada	27
Tecnologías	28
Docker	28
Componentes principales de Docker	28
Ventajas de utilizar Docker	30
Desventajas de utilizar Docker	31
Despliegue con Docker	32
Jmeter	33

Componentes principales de JMeter	34
Interfaz vs CLI	36
Ventajas de usar JMeter	36
Desventajas de JMeter	37
Pruebas de rendimiento con JMeter	38
Azure DevOps	44
Principales características de Azure DevOps	44
Ventajas de Azure DevOps	45
Seguridad	46
Arquitectura final con requerimientos de seguridad INTA	46
Modelo de datos	47
Resultado	49
Resultado obtenido	49
Trabajo futuro	51
Memoria del proyecto	52
Participación de la referente y parte interesada	52
Cumplimiento de objetivos	53
Objetivo principal	53
Objetivos específicos	53
Planificación original vs ejecución	54
Análisis FODA	57
Riesgos y Solventación	59
Análisis de etapas	60
Instalación del Servidor	60
Capacitación y determinación de requerimientos	60
Instalación física del servidor y pruebas de integración	61
Documentación de instalación	62
Capacitación de administradores	63
Instalación de los clientes	64
Instalación física en los nodos clientes y pruebas de rendimiento	65
Documentación de instalación	65
Documentación de uso/Manual de usuario	66
Capacitación de los usuarios	66
Documentación final	66
Bitácora de trabajo	67
Aporte del proyecto	69
Utilización de nuevas tecnologías	69
Aprendizaje	71
Resumen	75

Conclusiones	78
Apéndice	80
Glosario	80
Bibliografía	83
Anexos	84
Bitácora de trabajo	84
Cronología	84
Semana 1	84
Semanas 2-4	84
Semana 5	85
Semanas 6-9	85
Semanas 10-13	86
Semanas 14-15	87
Semanas 15-20	88

Agradecimientos

A mi familia y amigos, en especial a mis padres y hermanos, por su constante apoyo y motivación durante los distintos momentos vividos en este periodo.

A mis compañeros y amigos de la facultad, quienes hicieron de esta etapa no solo un proceso de aprendizaje, sino también una grata experiencia de la cual me llevo grandes recuerdos.

A los profesores que formaron parte de este camino de formación, por su manera de enseñar y por brindarnos el apoyo y las herramientas necesarias para insertarnos en el mundo profesional. Quiero hacer una mención especial a Fernando Genin, quien me convocó para realizar este proyecto, y a Adolfo Spinelli, que me brindaron la ayuda necesaria para llevarlo a cabo de manera exitosa.

A la referente funcional, Ariana Digilio, por depositar su confianza en mí. También quiero destacar su compromiso y predisposición para asegurar que este proyecto llegue a buen puerto.

Resumen

La subred de Recursos Fitogenéticos, un grupo dependiente del Instituto Nacional de Tecnología Agropecuaria (INTA), inició un proceso de informatización de los bancos de germoplasma que existen a lo largo del país. Para ello, estableció contacto con Crop Trust, empresa que posee el software GRIN Global, diseñado específicamente para este campo de estudio. Un banco de germoplasma es un lugar donde se conserva la diversidad genética de los cultivos, a través de la colecta, conservación, caracterización, evaluación, documentación y distribución de las poblaciones recolectadas.

Ante la falta de personal disponible dentro de su organización para llevar a cabo la implementación, la referente funcional, Ariana Digilio, se comunicó con la Facultad de Ingeniería en busca de ayuda. Para ello, se conformó un equipo liderado por profesionales de la Facultad y un estudiante alumno, quien tuvo la posibilidad de hacer de este proyecto su trabajo final de carrera.

El objetivo principal del proyecto fue implementar el software seleccionado en las instalaciones del INTA con el fin de informatizar y centralizar la información de los 21 bancos de germoplasma distribuidos por todo el país.

Para lograrlo, el proyecto incluyó diversas actividades, tales como el estudio y capacitación sobre la solución; la determinación de los requerimientos en términos de recursos y seguridad; el diseño de la arquitectura; la configuración y despliegue de los componentes de *software*; las pruebas de integración y rendimiento; la

documentación de instalación, administración y uso; y, finalmente, la capacitación de los usuarios que harán uso del sistema.

A pesar de los problemas, retrasos y errores, el resultado fue exitoso. Los usuarios ya utilizan el nuevo sistema, que les ayuda en sus tareas diarias. Toda la información obtenida durante las actividades es documentada en una base de datos especializada, registrando datos biológicos, coordenadas geográficas, fotografías, entre otros, generando información en diversos formatos.

Cabe destacar que el rol activo del estudiante en la toma de decisiones fue clave para alcanzar el objetivo. La interacción con los distintos actores involucrados en el proyecto, en sus distintos roles y funciones, también jugó un papel fundamental.

Este proyecto representó una gran oportunidad para aplicar nuevas tecnologías orientadas al agilismo y al mundo *cloud*. El desafío de trabajar con herramientas modernas, como la contenerización con **Docker**, las pruebas con **JMeter** y la integración con **Azure DevOps**, amplió significativamente la visión profesional. A través de este proceso, se lograron no solo los objetivos técnicos, sino también un crecimiento profesional. Se integraron principios DevOps y de gestión en la nube, lo que enriqueció la experiencia y el aprendizaje en un entorno dinámico y en constante evolución.

Introducción

La subred de Recursos Fitogenéticos carecía de un sistema informático para el almacenamiento y procesamiento de datos. Anteriormente, los datos se registraban en papel y hojas de cálculo, incluyendo las etiquetas de cada muestra, las observaciones, fotografías, entre otros. Además, el proceso para compartir y acceder a la información entre profesionales ubicados en diferentes localidades del país era ineficaz, ya que se debía hacer de forma manual, enviando archivos por email o compartiendo desde la nube.

Para ello, se ha encontrado una posible solución: una herramienta de *software* llamada GRIN Global Community Edition. Sin embargo, esta solución requería que alguien se capacite para lograr su instalación, el uso correspondiente y, posteriormente, capacite al resto del personal interesado.

La implementación de un producto de *software* para resolver un problema de información es una práctica habitual en la vida profesional de un ingeniero informático. Durante este proceso, lo que se destaca es su trabajo en la toma de decisiones a la hora de planificar, ejecutar y controlar la solución, tareas que requieren de las habilidades que debe poseer el ingeniero.

Objetivos del proyecto

El objetivo principal del proyecto es implementar el *software* seleccionado en la Red de Recursos Fitogenéticos perteneciente al INTA a fin de informatizar y centralizar la información de los bancos de germoplasma.

Del mismo surgen los siguientes objetivos específicos:

- Diseñar la arquitectura de la solución en los servidores del INTA y en los nodos clientes utilizados por los científicos.
- Implementar el *software* GRIN Global Community Edition,
- Capacitar al personal que hará uso del sistema.

Alcance del proyecto

El alcance del proyecto contempla la evaluación, planificación, implementación y la puesta en marcha de una base de datos para los bancos de germoplasma de INTA y un sistema para interactuar con los datos. Dichas actividades incluyen la generación de la correspondiente documentación, reuniones con el personal interesado y el representante del *software* para avanzar con las distintas etapas. Finalmente, se realizaron reuniones para capacitación del personal que hará uso de la solución.

Para dejar en evidencia las tareas realizadas se generó una minuta y un acta que fue firmada por las partes para dar su consentimiento y aprobación del trabajo ejecutado. Además, en la documentación se establecieron los límites entre las responsabilidades de cada uno de los actores. Una vez entregados los documentos,

se dará un periodo de soporte para dar un cierre final al proyecto. Luego de ese tiempo, la responsabilidad para mantener el *software* actualizado y resolución de problemas recaerá en el personal del INTA. Para ello, los documentos contienen toda la información necesaria para lograr comprender el funcionamiento de la solución y también, podrán reversionar con la información adicional que consideren pertinente.

Soporte

Luego de la finalización del proyecto, se acordó con la referente funcional estar disponible para brindar soporte durante el primer mes de puesta en producción. Se mostrará predisposición a responder cualquier consulta que pueda tener la referente funcional o cualquier usuario del sistema, proporcionando orientación y resolución de problemas según sea necesario.

Impacto

La ejecución del proyecto busca facilitar la realización de las tareas diarias del personal y reducir el tiempo dedicado a tareas operativas, con el fin de poder enfocarse más en sus actividades. Toda la información obtenida durante las actividades será documentada en una base de datos especializada, registrando datos biológicos, coordenadas geográficas, fotografías, entre otros, generando información en diversos formatos. Este sistema permitirá informatizar los procesos actuales, centralizando el acceso a la información para los 21 bancos de germoplasma distribuidos en el país.

Dada la naturaleza del proyecto, el desarrollo del *software*, en particular, no representa un aspecto innovador. Sin embargo, la implementación de un banco de germoplasma de estas características es algo nuevo localmente, dado que no existe uno de esta envergadura en todo el país. El factor innovador estará vinculado con la promoción de un entorno de trabajo más colaborativo, mediante la conexión de todas las estaciones de trabajo de INTA, optimizando los procesos laborales y estableciendo una fuente única de información accesible para todos.

Cabe destacar que, si bien GGCE funcionará de manera centralizada, la entrada de datos se llevará a cabo por cada uno de los integrantes de la subred de Recursos Fitogenéticos.

Entregables

Los entregables del proyecto son:

- Documentación de instalación de servidor GGCE
- Documentación de administración de servidor GGCE
- Documentación de instalación y uso de cliente Curator Tool (Herramienta de escritorio)
- Documentación de uso del sitio web GGCE.

Análisis del problema

Dominio del problema

Los bancos de germoplasma son espacios de conservación y manejo de la diversidad genética de especies vegetales y animales, especialmente aquellas utilizadas para la alimentación, la agricultura, la horticultura, la silvicultura, y otros usos relacionados. Un banco de germoplasma es una colección de muestras biológicas, principalmente semillas, esporas, cultivos in vitro y otros materiales reproductivos (como embriones o tejidos vegetales), que se almacenan de manera controlada con el fin de preservar la variabilidad genética de las distintas especies para su uso futuro.

Los bancos de germoplasma son fundamentales para la conservación de la biodiversidad y el desarrollo sostenible en áreas como la agricultura y la biotecnología. El objetivo es garantizar que el material genético de especies vegetales y animales se conserve a largo plazo, para poder recuperarlo y utilizarlo en caso de que las especies originales se vean amenazadas por factores como el cambio climático, la pérdida de hábitat, enfermedades o pérdida de diversidad genética.

Para ello, los investigadores deben documentar y etiquetar todas las muestras con las que el banco cuenta. Este registro es necesario para poder cumplir con los objetivos y mantener el material de manera organizada.

Actores del sistema

- Demandante del Proyecto: Instituto Nacional de Tecnología Agropecuaria (INTA).
- Red de Recursos Fitogenéticos, grupo dependiente del INTA(Referentes funcionales).
 - Bubillo, Rosana
 - Digilio, Ariana
 - Defacio, Raquel
- Área de gerencia del INTA
 - Sutz Kaelin, Guillermo Gabriel
- Responsable de cuentas en INTA
 - Juan, Cristian Oscar
- Responsable de infraestructura en INTA
 - Zabal, Juan Pablo
- Responsable de ciberseguridad en INTA
 - Herrera, Hernán Maximiliano
- El equipo, en representación de la Facultad de Ingeniería UNMDP, conformado por los directores y el estudiante.
 - Escalante, Santiago (Estudiante)
 - Genin, Fernando (Director)
 - Spinelli, Adolfo (Director)
- Representante del *software* GRIN Global
 - Alarcón, Juan Carlos

Problema a Resolver

La subred de Recursos Fitogenéticos carece de un sistema informático para el almacenamiento y procesamiento de datos. Actualmente, los datos se registran en papel y hojas de cálculo, incluyendo las etiquetas de cada muestra, las observaciones, fotografías, entre otros. Además, el proceso para compartir y acceder a la información entre profesionales ubicados en diferentes localidades del país es ineficaz, ya que se debe hacer de forma manual, enviando archivos por email o compartiendo información desde la nube.

Ante esto, se ha encontrado una posible solución: una herramienta de *software* llamada GRIN Global Community Edition. Sin embargo, esta solución requiere que alguien se capacite para su instalación y, posteriormente, capacite al resto del personal interesado.

Para lograrlo, se firmó un convenio de cooperación técnica a fin de evaluar las necesidades de *hardware* y *software* con el objetivo de implementar GGCE en los bancos de germoplasma. Los recursos de *hardware* y parte del *software* serán provistos por el INTA, Crop Trust aportará GGCE, mientras que el know-how para el despliegue de aplicaciones utilizando herramientas de contenedorización, así como el entendimiento y la capacitación en el uso del *software*, será proporcionado por el estudiante.

Aporte del proyecto

Parte interesada

Desde hace más de un año, las personas pertenecientes al grupo de recursos fitogenéticos, un grupo dependiente del INTA, han estado en contacto con la empresa Crop Trust participando de diferentes eventos y charlas informativas acerca del sistema para bancos de germoplasma GRIN Global Community Edition(GGCE).

En este caso en particular, los investigadores observaron la oportunidad de poder aplicar la solución en sus instalaciones y comenzar hacer uso de este sistema más eficiente. Para ello, necesitaban contar con una persona con los conocimientos necesarios para llevar a cabo todo el proceso de implementación y que pudiera dar apoyo en el proceso de capacitación de la herramienta.

El equipo

El equipo está conformado por un único estudiante, un director y un co-director, mencionados anteriormente, y la referente funcional, la Ingeniera Ariana Digilio. Desde su rol ha aportado su conocimiento en el campo de la genética lo cual ha ayudado a comprender el dominio y , con ello, el problema.

El vínculo con el director y el codirector se estableció durante los años anteriores de la carrera y esto ha facilitado la comunicación con ellos a la hora de transmitir los resultados, avances, dudas o posibles problemas. Siempre se ha alentado o buscado la reflexión del estudiante a la hora de entender el qué y el cómo resolver los problemas.

La razón por la cual se optó por un equipo conformado por un estudiante es que no calificaba en términos de horas requeridas para un Proyecto Final de más de una persona.

El no tener otro estudiante en el equipo puede que haya repercutido en las decisiones tomadas, aunque la relación, el apoyo brindado y la predisposición por cada uno de los miembros ha sido un factor fundamental para poder cumplir con todo lo propuesto.

Análisis FODA

A través del análisis FODA se identificaron las fortalezas y oportunidades que pueden favorecer al equipo y, en contraparte, las debilidades y amenazas que podrían perjudicarlo en el transcurso del proyecto. Los puntos que integran el análisis son:

Fortalezas

- Conocimientos en bases de datos y herramientas de contenerización.
- Experiencia en trabajos de desarrollo realizados a lo largo de la carrera.
- La guía de los profesionales involucrados en este proyecto.

Oportunidades

- Adquirir la experiencia en un proyecto de infraestructura y despliegue de soluciones para etapa de producción.

Debilidades

- Poca experiencia en la toma de decisiones a nivel profesional.
- Poca experiencia a la hora de estimar el tiempo del proyecto.
- Poco conocimiento sobre el campo de interés.
- Dependencia de los tiempos de respuesta de las entidades involucradas para consultas y validación de avances que demoran el proyecto.

Amenazas

- El demandante se decida por otra oferta.
- La solución la lleve a cabo otro equipo de profesionales.
- El contexto económico actual y los recursos con los que cuenta el demandante.

Análisis de Riesgos

En esta sección se presentan algunos de los riesgos identificados en el análisis FODA anterior, los cuales podrían afectar la continuidad o el desarrollo del proyecto. Estos riesgos han sido evaluados y etiquetados según sus respectivas probabilidades de ocurrencia y la gravedad en caso de que se materialicen. Las probabilidades y la gravedad se califican en una escala del 1 al 3, siendo 1 bajo y 3 Alto. El producto de ambas variables determinará el peso de cada riesgo, con una escala de valores que va del 1 al 9. Si el valor del peso calculado es 6 o mayor, se ha elaborado un plan de contingencia para reducir su gravedad en caso de que suceda.

Riesgo	Descripción	Consecuencia	Probabilidad	Gravedad	Peso
R01	El integrante abandona el proyecto	La implementación se retrasa hasta que se encuentre otro estudiante dispuesto a tomar el lugar.	1	3	3
R02	Imposibilidad de coordinar reuniones con los referentes funcionales	Imposibilidad de determinar los requerimientos, retrasos en el cronograma.	2	3	6
R03	Ausencia temporal del integrante	Retrasos en el cronograma	3	2	6
R04	El proyecto lo lleve a cabo otro equipo de profesionales	El trabajo final queda sin concluir.	1	3	3
R05	El proyecto no pueda solventarse	El trabajo final concluye sin poder completarse o se	2	3	6

	con los recursos que se cuentan	retrasa hasta contar con los recursos necesarios			
R06	La solución no cumpla con los requerimientos de seguridad de la institución.	El trabajo final concluye sin poder completarse.	2	3	6
R07	Falta de documentación	Dificulta el entendimiento de la solución. Retrasos en el cronograma.	1	2	2

Tabla 1. Análisis de Riesgos del Proyecto

Medidas de contingencia

Para mitigar los riesgos, se establecieron los siguientes planes de contingencia:

- **R02:** Ante la imposibilidad de coordinar con la referente funcional y otras partes involucradas en la realización de reuniones sincrónicas para avanzar con el desarrollo del proyecto, se tienen en cuenta la comunicación a través de canales alternativos asincrónicos, como correos electrónicos, servicios de mensajería, entre otros. El interés y el compromiso de las partes son fundamentales para continuar y evitar retrasos inesperados.
- **R03:** Ante la ausencia del estudiante durante el periodo de realización del proyecto, se ha decidido realizar horas extras para cumplir con los plazos establecidos.
- **R05:** En caso de que el demandante no cuente con los recursos necesarios para solventar el proyecto, se implementará una prueba *on-premise* con los recursos del integrante. Esto permitirá continuar con las tareas mientras se espera por los recursos necesarios y mitigar posibles retrasos en el cronograma.
- **R06:** Para este caso se deberá agregar una capa adicional de seguridad a la solución con el fin de cumplir con los requerimientos de seguridad y poder llevar a cabo su implementación.

Estimación inicial

Entre las debilidades enunciadas en el análisis FODA, se menciona la falta de experiencia a la hora de estimar los tiempos del proyecto. Por lo que dichas estimaciones pueden tener, en ciertos casos, desvíos considerables. Se dieron una serie de reuniones con los directores del proyecto que permitieron hacer un acercamiento en la determinación de las tareas necesarias y el alcance del proyecto.

El proyecto fue dividido por componentes de *Software*. La solución sigue una arquitectura o modelo de cliente servidor, por lo que se tomó como eje cada producto y se subdividió en etapas, compuestas por sus tareas correspondientes. Al dividir las etapas en tareas más sencillas, fue posible obtener una estimación lógica y aceptable.

El diagrama de Gantt del proyecto contempla la instalación del servidor y la instalación de los clientes, la división por componentes como se mencionó anteriormente . Las etapas que pertenecen a cada una son las siguientes:

- **Instalación del servidor**
 - Capacitación y determinación de requerimientos
 - Instalación física del servidor y pruebas de integración
 - Documentación de instalación
 - Capacitación de administradores
- **Instalacion de los clientes**
 - Instalación física en los nodos clientes y pruebas de rendimiento

Implementación del sistema GRIN Global Community Edition para la gestión de los bancos de germoplasmas de la Red de Recursos Fitogenéticos del INTA

Proyecto Final de Grado - Escalante, Santiago

- Documentación de instalación
- Documentación de uso/Manual de usuario
- Capacitación de los usuarios

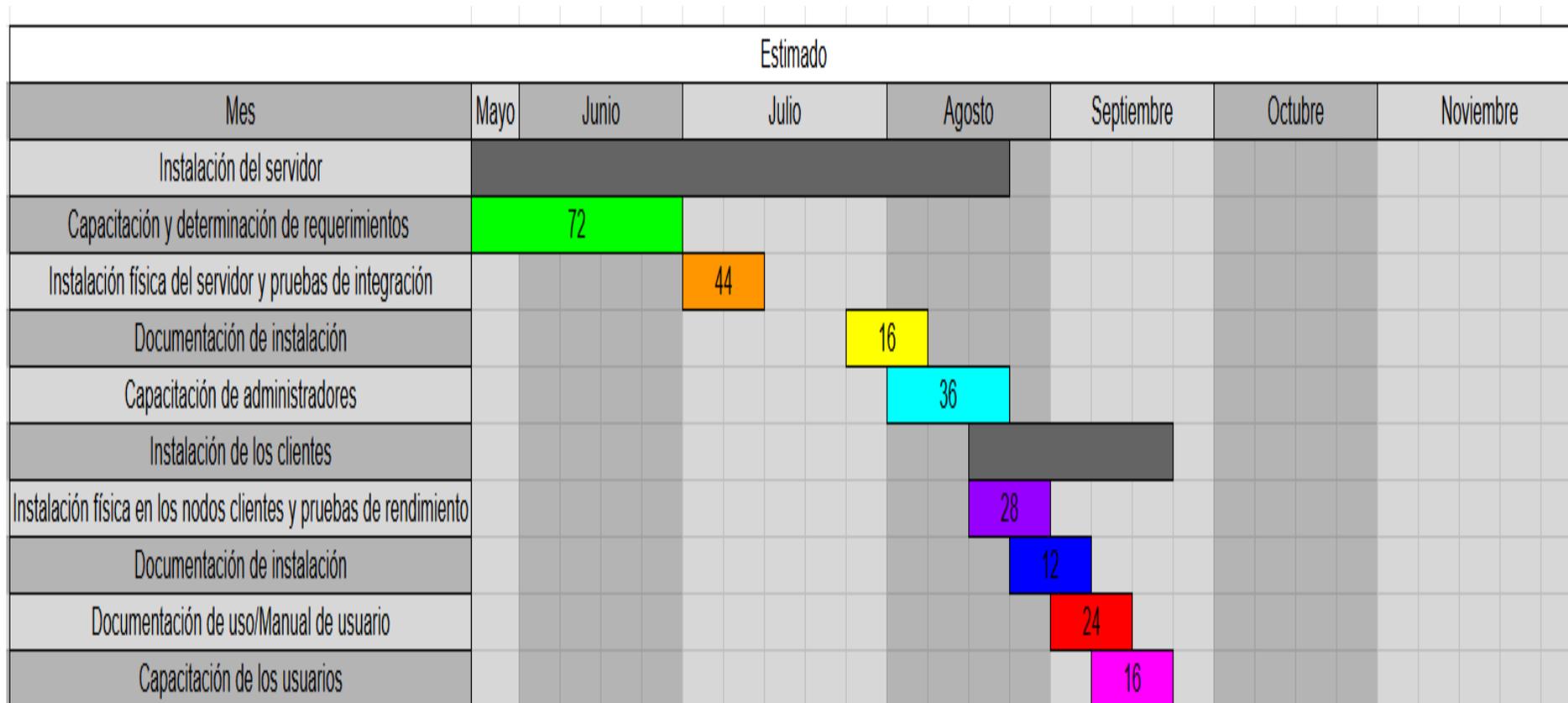


Figura 1. Diagrama de Gantt y cronograma estimado.

La planificación se estimó para llevarse a cabo de manera secuencial de modo que cada tarea comience una vez finalizada la tarea anterior, sin tener en cuenta la posibilidad de realizar tareas en paralelo. Aunque no se presentó como una etapa propia del proyecto, es importante destacar que se llevó la Bitácora de Trabajo de principio a fin, tomando nota de los datos y acontecimientos relevantes en cada una de las etapas del trabajo. Otra etapa que no se tuvo en cuenta a la hora de estimar inicialmente es la documentación final, que se desarrollará en la sección [Planificación original vs ejecución](#).

Metodologías

Metodología de trabajo

Para el desarrollo del proyecto, se adoptó una metodología de trabajo en cascada con entregas parciales. Las tareas se llevaron a cabo de forma secuencial, avanzando a la siguiente una vez completada la anterior. Si bien este esquema tiene sus desventajas frente a los esquemas ágiles, debido a que no se mantiene una retroalimentación constante con el cliente, permitió seguir una estructura clara frente a la falta de experiencia del equipo en el ámbito genético, algunas de las tecnologías a utilizar y la solución de *software* a implementar.

Se realizó una entrega parcial del sistema para obtener la validación del cliente y permitirle comenzar a utilizar la solución. De esta manera, el cliente pudo familiarizarse con la herramienta, interactuando con el sistema y consultando la

documentación brindada para entender el modelo de datos y los flujos básicos antes de recibir toda la funcionalidad de la solución.

De las etapas definidas del proyecto, la etapa de instalación de los clientes se postergó hasta tener los conocimientos suficientes sobre el sistema y el dominio del problema. Esto permitió alcanzar la madurez necesaria para poder resolver las consultas por parte de las partes interesadas.

Metodología de implementación

La implementación del sistema no pudo comenzar hasta recibir las credenciales para acceder al servidor por parte del cliente. Mientras tanto, se decidió continuar con el estudio de las tecnologías y de la solución para lograr de manera eficiente la instalación, sin imprevistos. Cabe destacar que se optó por una solución basada en contenedores, más eficiente en el consumo de recursos y portable, lo que evitó limitaciones. Una vez recibidas las credenciales, se probaron los accesos. Tras acceder exitosamente, se comenzó con la instalación paso por paso.

Primero, se desplegó el contenedor con la base de datos. Una vez configurada, se probó el acceso con el usuario administrador y se realizó una consulta de prueba. Luego, se configuró el archivo *docker-compose.yaml* para el despliegue de la solución y la correcta integración con el contenedor de base de datos. En este proceso, se necesitó apoyo del representante del *software*, lo que ocasionó un ligero retraso, que se compensó gracias a la preparación previa sobre el uso de la herramienta Docker. Una vez validado el correcto funcionamiento, se informó al equipo y se coordinó una reunión con la referente funcional y las partes interesadas.

Como se mencionó anteriormente, la instalación de los clientes se hizo casi al final del proyecto. Para las pruebas de rendimiento, no fue necesario esperar hasta esa fecha, ya que se instaló en la máquina del estudiante. Esta decisión se tomó en función de la necesidad de contar mayor experiencia en el campo y un mayor entendimiento del funcionamiento del sistema, a fin de tener la madurez necesaria para poder responder consultas y realizar demostraciones.

Metodología de pruebas

Se llevaron a cabo pruebas de integración y rendimiento sobre el sistema. El objetivo de estas pruebas fue validar que la implementación fue exitosa y el funcionamiento es el esperado. Además, se evaluó la capacidad del sistema para responder a escenarios límites.

Las pruebas de integración tuvieron lugar una vez finalizada la instalación del servidor GGCE y tras instalar el cliente Curator Tool en la máquina del estudiante. En la herramienta, se configuró el dominio de la *API* del servidor registrado y se probó la conexión, obteniendo una respuesta exitosa. Además de verificar la conexión entre el cliente y el servidor, estas pruebas incluyeron en operaciones como iniciar sesión, buscar registros, entre otras acciones.

Las pruebas de rendimiento se basaron en simular múltiples peticiones al dominio en un periodo corto de tiempo para evaluar la capacidad del sistema de responder a una cantidad considerable de solicitudes. Para ello, se utilizó la herramienta JMeter, que resultó ser de fácil entendimiento y muy intuitiva.

Implementación del sistema GRIN Global Community Edition para la gestión de los bancos de germoplasmas de la Red de Recursos Fitogenéticos del INTA
Proyecto Final de Grado - Escalante, Santiago

En la sección [pruebas de rendimiento con JMeter](#) se detalla más acerca de las pruebas realizadas.

Diseño del sistema

Arquitectura

Arquitectura simplificada

A continuación, se presenta un diagrama que muestra los componentes principales del sistema y cómo se encuentran relacionados entre sí. La arquitectura general sigue un modelo de cliente-servidor, en el que los clientes son la aplicación web y la aplicación de escritorio. Por otro lado, tenemos la *API* que utilizan ambos clientes para comunicarse y consumir datos provenientes de la base de datos de la solución. Cabe destacar que el cliente instalado en los nodos del personal se subdivide en 2 herramientas: *Curator Tool* y *Search Tool* que cumplen distintas funciones pero se comunican entre sí. Para obtener o enviar datos a la base de datos, *Search Tool* es responsable de la comunicación directa con la *API*, mientras que *Curator Tool* es una herramienta de edición y visualización de los datos.

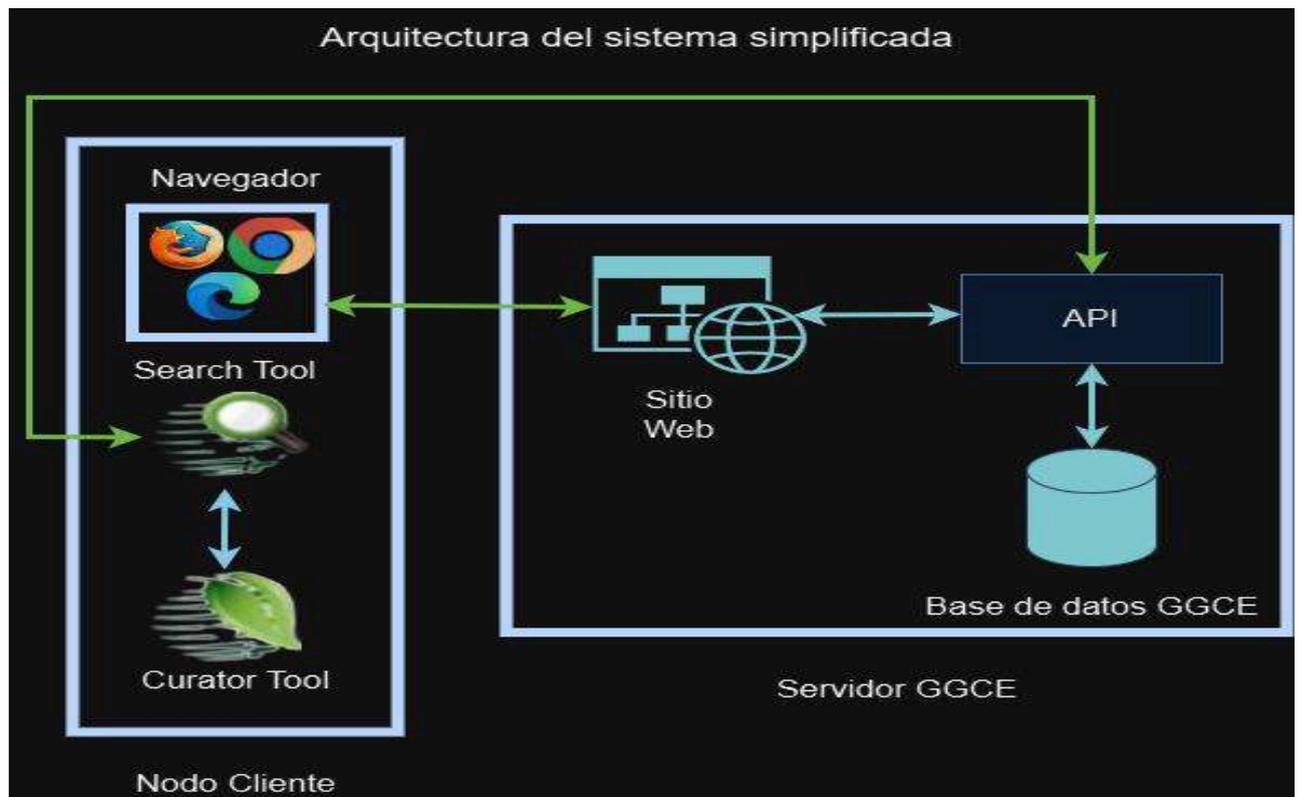


Figura 2. Arquitectura simplificada de la solución.

Tecnologías

Docker

Docker es una plataforma de *software* que permite crear, desplegar y ejecutar aplicaciones en contenedores. Docker simplifica el desarrollo y la gestión de aplicaciones, permitiendo a los equipos ser más ágiles y productivos.

Componentes principales de Docker

- **Contenedores:** son unidades ligeras y portátiles que encapsulan la aplicación y todas sus dependencias (librerías, configuraciones, etc) para que pueda ejecutarse de manera consistente en cualquier entorno. Los contenedores son más eficientes que las máquinas virtuales ya que

comparten el mismo núcleo del sistema operativo, lo que permite un mejor uso de los recursos.

- **Imagen:** Una imagen de Docker es un conjunto de archivos y configuraciones que incluyen el código fuente de la aplicación, dependencias, bibliotecas y otras configuraciones necesarias para ejecutar la aplicación. Las imágenes son inmutables y pueden ser versionadas. Es como el archivo binario de una aplicación. Al ejecutarse, se crea un contenedor con todo lo definido en la imagen.
- **Dockerfile:** Es un archivo que contiene una serie de instrucciones que Docker sigue para crear una imagen. En el Dockerfile se definen todos los pasos necesarios, como la instalación de dependencias, configuración de variables de entorno, copiado de archivos, etc. Es la receta para crear imágenes de manera automatizada.
- **Docker Registry:** es un sistema para almacenar y distribuir imágenes Docker. Docker Hub es el registro más popular, pero también es posible crear tu propio registro privado, en tu infraestructura local o en la nube (Azure Container Registry, por ejemplo). Esto permite centralizar el acceso a las imágenes confiables y verificadas, recordemos que no dejan de ser programas por lo que sigue existiendo el riesgo por malware.

- **Docker Compose:** es una herramienta que permite definir y ejecutar aplicaciones multicontenedor. Usando un archivo YAML, es posible definir cómo se deben construir y configurar múltiples contenedores, establecer redes entre ellos y gestionar volúmenes de almacenamiento. Esto simplifica la administración de aplicaciones complejas que necesitan varios servicios.
- **Docker Engine:** es el motor que ejecuta y gestiona los contenedores.
- **Volúmenes:** Son una forma de persistir los datos fuera de los contenedores. Dado que los contenedores son efímeros, cualquier dato almacenado directamente en un contenedor se pierde cuando se elimina. Los volúmenes permiten almacenar datos de manera persistente, de modo que se pueden reutilizar entre contenedores y mantener la integridad de los datos. Para persistir los datos se debe crear el volumen, asociarlo al contenedor que contiene datos a persistir, mapear el directorio donde se encuentra los archivos en el contenedor con el directorio de la máquina destino.
- **Redes:** Los contenedores pueden comunicarse entre sí mediante redes virtuales, lo que permite a las aplicaciones distribuidas o servicios interactuar entre sí de manera segura.

Ventajas de utilizar Docker

Utilizar este enfoque para el despliegue de aplicaciones tiene las siguiente ventajas:

- **Portabilidad:** Los contenedores Docker pueden ejecutarse en cualquier lugar, desde la máquina del desarrollador hasta el servidor en la nube, sin importar el sistema operativo.
- **Aislamiento:** Cada contenedor opera de manera independiente, lo que evita conflictos entre aplicaciones y facilita la gestión de recursos.
- **Escalabilidad:** Docker permite escalar aplicaciones fácilmente, ya que se pueden crear y destruir contenedores rápidamente según la demanda.
- **Eficiencia:** A diferencia de las máquinas virtuales, los contenedores comparten el mismo núcleo del sistema operativo, lo que los hace más ligeros y rápidos en términos de inicio y uso de recursos.
- **Facilidad de integración:** Docker se integra bien con herramientas de CI/CD, facilitando la automatización del desarrollo, pruebas y despliegue de aplicaciones.

Desventajas de utilizar Docker

Entre las desventajas de utilizar Docker tenemos se encuentran las siguientes:

- **Curva de aprendizaje:** Para aquellos nuevos en la contenedorización, puede haber una curva de aprendizaje al entender conceptos como imágenes, contenedores, volúmenes y redes.

- **Complejidad de gestión:** A medida que se despliegan más contenedores, la gestión y orquestación pueden volverse complicadas, especialmente sin herramientas adecuadas como Kubernetes.
- **Persistencia de datos:** La gestión de datos persistentes puede ser un desafío, ya que los contenedores son efímeros por naturaleza. Se requiere una planificación cuidadosa para almacenar datos de manera segura. Los volúmenes en Docker son una forma de persistir los datos.

En nuestro caso, contamos con experiencia y conocimientos sobre la herramienta. Además, se han estimado los tiempos considerando la posibilidad de aprender todos los detalles de su uso como parte del proyecto. Por esta razón, las desventajas enunciadas no representan un problema para este escenario.

Despliegue con Docker

Se observó que existía una solución basada en contenedores para GGCE, una tecnología innovadora y eficiente por lo que se decidió implementar. La solución incluye el sitio web y la *API*, y además recomienda una solución contenerizada para la base de datos, aunque es posible utilizar la base de datos propia de la institución o una diferente que no esté basada en contenedores. A continuación, se presenta la arquitectura definida previamente, pero con más detalle sobre el servidor, donde se puede observar los componentes y los contenedores necesarios para su implementación.

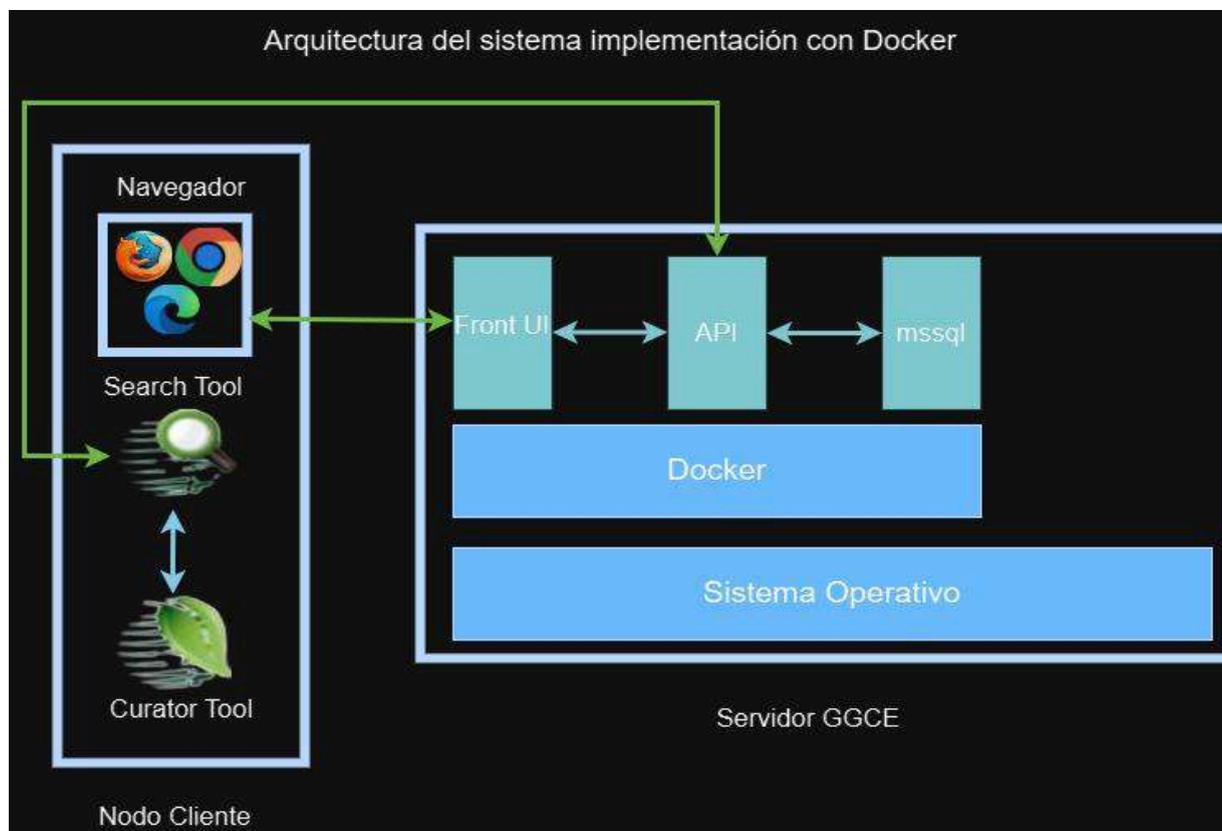


Figura 3. Arquitectura de la solución GGCE en su versión basada en contenedores.

Jmeter

Apache JMeter es una herramienta de código abierto diseñada principalmente para realizar pruebas de rendimiento y pruebas de carga en aplicaciones web, pero también es posible llevar a cabo otros tipos de pruebas, como pruebas funcionales y pruebas de estrés. JMeter se utiliza para medir la capacidad de un sistema para manejar el tráfico generado por los usuarios y simular diversos escenarios de carga, por ejemplo, en picos de demanda, lo que ayuda a identificar cuellos de botella, rendimientos por debajo del objetivo y problemas en la infraestructura de servidores, bases de datos, aplicaciones, en otros.

Apache JMeter es una aplicación basada en Java que permite simular una cantidad considerable de usuarios interactuando con una aplicación web o un servidor y medir el rendimiento bajo diferentes condiciones de carga. Su capacidad para simular tráfico HTTP(s) y otros protocolos (como FTP, JDBC, JMS, etc) lo convierte en una herramienta muy versátil para realizar pruebas de rendimiento.

Aunque fue inicialmente diseñada para probar aplicaciones web, con el tiempo se ha ampliado para cubrir una amplia gama de pruebas, incluidas pruebas de servicios web (SOAP, REST), bases de datos (JDBC), aplicaciones FTP, y más.

Componentes principales de JMeter

Los principales componentes de JMeter son:

- **Elementos de Prueba**

Son los componentes básicos que forman una prueba.

- **Test Plan(Plan de prueba):** es el contenedor principal donde se define la configuración de las pruebas a realizar. Contiene todos los elementos (como thread groups, samplers, listeners, etc).
- **Thread Groups(Grupo de Hilos):** representa el número de usuarios virtuales a simular. Cada hilo del grupo representa un usuario virtual. Se puede especificar el número de hilos, el tiempo de inicio, el tiempo entre solicitudes, y cuantas veces se repetirá la prueba (loop)

- **Samplers:** son los elementos que realmente representan las solicitudes o interacciones con el sistema. Existen diferentes tipos según el protocolo utilizado: HTTP Request, FTP Request, JDBC Request, SOAP/REST Web Service Request.
- **Listeners:** Recopilan y muestran los resultados de las pruebas. Existen diferentes listeners que permiten ver los resultados en forma de gráficos, tablas y logs.
- **Timers:** el tiempo entre solicitudes. Esto es útil para simular el comportamiento de usuarios reales que no hacen solicitudes simultáneas. Existen Constant Timers(tiempo fijo) y Gaussian Random Timer(tiempo aleatorio en un rango definido).
- **Assertion:** verifican que las respuestas de las solicitudes sean correctas y estén dentro de los parámetros esperados. Esto es útil para medir la consistencia de los datos y el comportamiento del sistema durante las pruebas. Existen Response Assertion que verifica si el contenido de la respuesta es correcto y Duration Assertion que verifica si la respuesta se realiza dentro de un tiempo definido.
- **Variables:** Permiten almacenar valores, como parámetros de usuario, que luego pueden ser utilizados en diferentes partes del script de prueba.

- **Funciones:** Permiten obtener información dinámica durante la ejecución de las pruebas, como la fecha y hora actual, valores aleatorios, o resultados de variables previas.

Interfaz vs CLI

JMeter tiene una estructura modular que permite a los usuarios configurar, personalizar y ejecutar pruebas de rendimiento a través de una interfaz gráfica o mediante scripts.

- **Interfaz Gráfica (GUI):** JMeter proporciona una interfaz gráfica fácil de usar para crear y ejecutar pruebas. Es muy útil durante el desarrollo de pruebas y para ver resultados detallados en tiempo real. Sin embargo, la **GUI** puede ser limitada cuando se realizan pruebas a gran escala o en entornos de producción.
- **Interfaz de Línea de Comandos (CLI):** JMeter también puede ejecutarse en modo **CLI** (línea de comandos), lo cual es ideal para ejecutar grandes pruebas de carga en entornos de servidor, ya que no tiene la sobrecarga gráfica. De este modo, los resultados se pueden generar en archivos de texto, lo que facilita su análisis.

Ventajas de usar JMeter

- **Código Abierto y Gratuito:** JMeter es una herramienta **open-source**, lo que significa que es completamente gratuita y tiene una gran comunidad que contribuye al proyecto.
- **Soporte para Diferentes Protocolos:** Soporta una amplia variedad de protocolos, incluyendo HTTP, HTTPS, FTP, JMS, JDBC, SOAP, REST, y más.
- **Escalabilidad:** Puedes aumentar el número de hilos (usuarios virtuales) de manera eficiente, lo que lo convierte en una herramienta adecuada tanto para pruebas pequeñas como para pruebas de gran escala.
- **Flexible y Extensible:** JMeter permite extender su funcionalidad mediante la adición de plugins y personalización con funciones adicionales.
- **Interfaz Visual:** La interfaz gráfica facilita la creación, configuración y ejecución de pruebas.

Desventajas de JMeter

- **Uso de Memoria en Pruebas a Gran Escala:** JMeter es una herramienta basada en Java, por lo que puede consumir una cantidad considerable de recursos del sistema cuando se realizan pruebas a gran escala.

- **Curva de Aprendizaje:** Aunque es relativamente fácil de usar para principiantes, la complejidad aumenta cuando se requieren configuraciones avanzadas, lo que puede presentar una curva de aprendizaje.
- **Requiere Conocimientos Técnicos:** Para exprimir todo su potencial, se necesitan conocimientos básicos de pruebas de rendimiento y algunas habilidades técnicas para configurar las pruebas correctamente.

En el caso del proyecto, inicialmente se intentó utilizar la CLI desde un contenedor con una imagen de JMeter, pero en reiteradas ocasiones se presentaron errores en el archivo del plan de pruebas generado manualmente. Sin poder dar una solución al mismo, se decidió utilizar la interfaz gráfica, la cual fue mucho más intuitiva e incluso genera el archivo con el plan de pruebas de forma automática. Al ser una aplicación que utilizan no más de 40 personas, no requiere realizar grandes pruebas de carga. Por lo tanto, con los recursos disponibles y la opción de la GUI fue suficiente para llevarlas a cabo.

Pruebas de rendimiento con JMeter

Las pruebas de rendimiento tuvieron como objetivo medir la capacidad del sistema de responder de manera exitosa a una cantidad considerable de peticiones, simulando la interacción de los usuarios en un periodo determinado de tiempo. Para ello, se realizaron dos pruebas:

En primera instancia, se probó con 50 solicitudes de usuarios en 5 segundos, lo que dió un promedio de 10 usuarios por segundo. Para este escenario, las pruebas

fueron exitosas, obteniéndose una respuesta con el *código 200 (OK)* para todas las solicitudes.

La siguiente prueba consistió en duplicar la cantidad de usuarios. En este caso, se presentaron 20 solicitudes por segundo y el resultado también fue exitoso. Considerando que estos números superan casi el doble del número total de usuarios que tendrá el sistema, siendo de uso interno y exclusivo para quienes tengan acceso a la red del INTA, no se continuó con pruebas con mayor cantidad de solicitudes ya que se consideró que eran suficientes.

Además de la cantidad de solicitudes exitosas, también se evaluó la latencia, es decir, el tiempo en el que tarda en responder el servidor a una solicitud.

A continuación, se presenta un resumen de los resultados de las pruebas. Se deberán omitir los errores, ya que fueron causados por problemas iniciales al comenzar a utilizar la herramienta JMeter y no son representativos del comportamiento real del sistema. También se incluirá un gráfico que muestra la latencia de las solicitudes. Las primeras solicitudes terminaron con un *timeout* debido a no poder establecer conexión con el servidor, como se mencionó anteriormente.

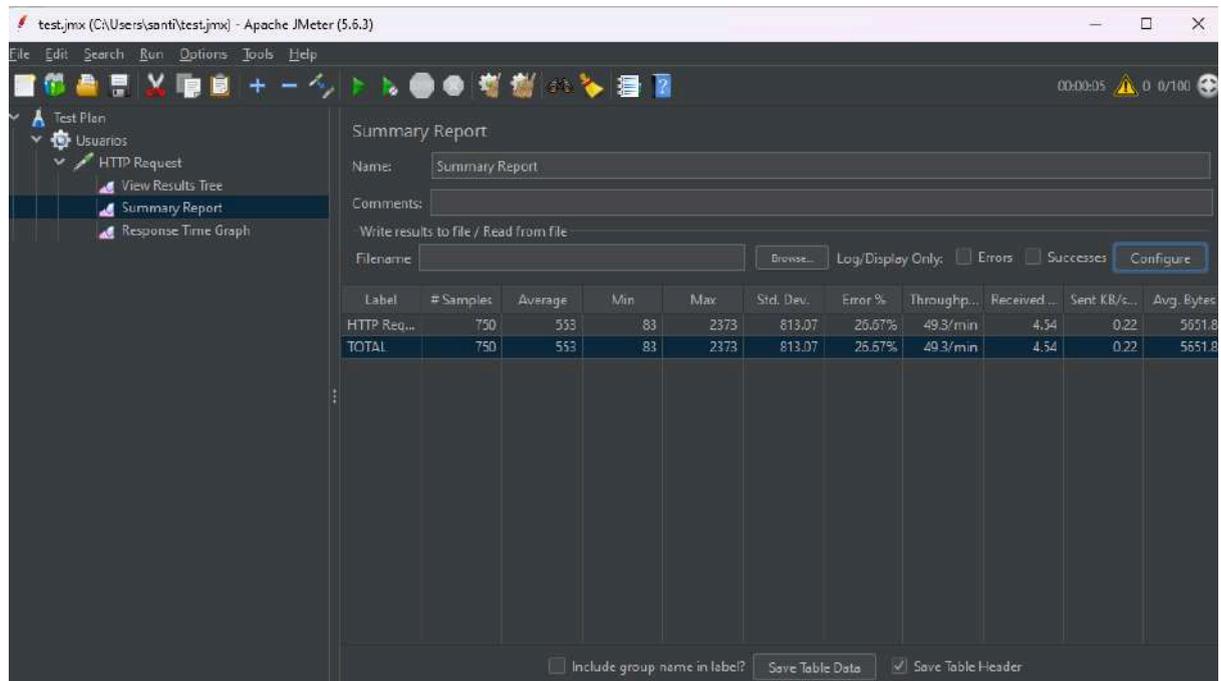


Figura 4. Resumen de las pruebas de rendimiento.

Aquí se destacan los siguientes campos:

- **Muestras(Samples):** se realizaron 750 peticiones en total para ambas pruebas. En 7 tandas de 50 solicitudes en 5 segundos para la primera prueba y 4 tandas de 100 solicitudes en 5 segundos para la segunda prueba.
- **Promedio(Average):** es el tiempo de respuesta promedio de las 750 peticiones, el valor obtenido es de 553 milisegundos.
- **Min:** el tiempo de respuesta más bajo a una solicitud. En este caso 83 milisegundos.

- **Max:** el tiempo de respuesta más alto a una solicitud. En este caso 2373 milisegundos y se debe a los timeouts mencionados anteriormente.
- **Desviación Estándar(Std.Dev):** es la variación entre los valores de respuesta obtenidos. Un valor alto indica que existe una diferencia considerable entre los tiempos de respuesta bajos y altos. En nuestro caso, la desviación es de 813.07 ms, por lo que existe una variación considerable entre los valores. Esto es consistente con lo ocurrido en las primeras 200 solicitudes, que terminaron en timeout, y las 500 restantes, que fueron exitosas.
- **Porcentaje de error:** representa las solicitudes que fallaron sobre el total de las solicitudes realizadas. El valor es de 26,67% de 750 solicitudes, indicándonos que fallaron 200 solicitudes.
- **Tasa de transferencia(Throughput):** representa la tasa de transacciones por minuto o peticiones completadas por segundo. En este caso, el valor es de 49,3 transacciones por minuto, lo que equivale a 0,82 peticiones completadas por segundo.

Analizando los valores obtenidos, concluimos que los errores iniciales afectaron las métricas generales. Sin embargo, nos permiten deducir que la latencia en condiciones normales se encuentra en un rango de 100-300 ms, la cual es considerado aceptable. Además, el valor mínimo supera dicho rango, lo que indica

una latencia ideal. Al no tratarse de un sistema en tiempo real, como aplicaciones de streaming o videojuegos que requieren latencias muy bajas, se concluye que las pruebas han sido exitosas.

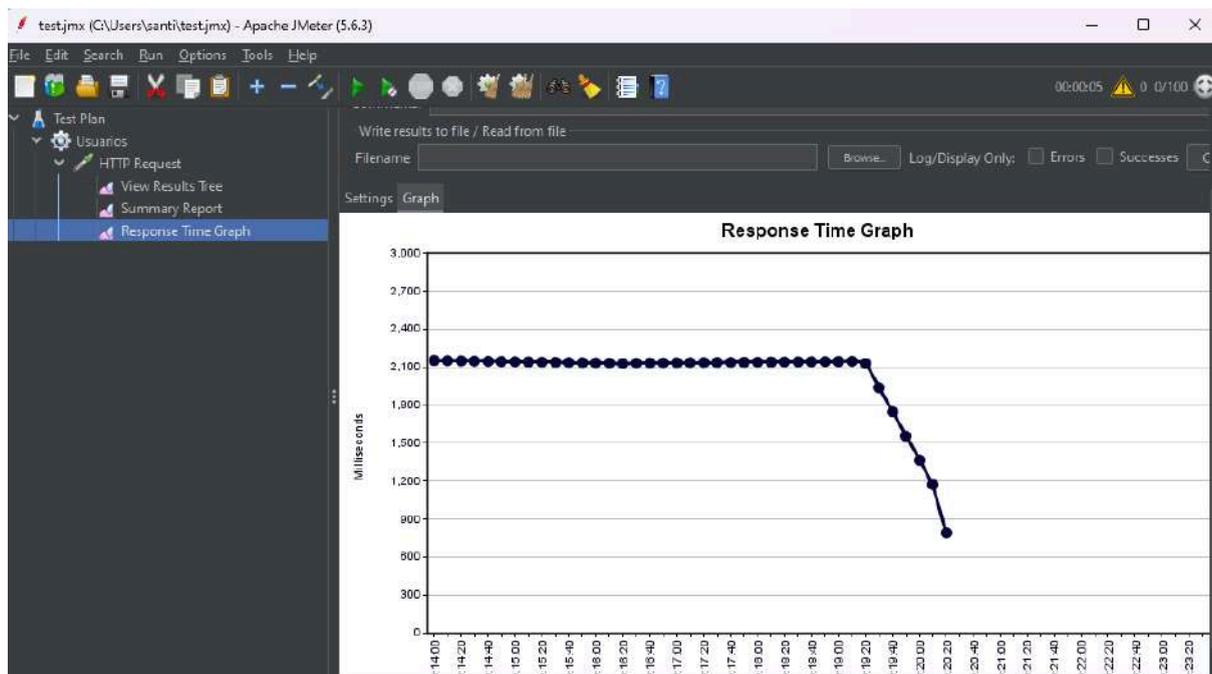


Figura 5. Gráfico del tiempo de respuesta promedio por tiempo.

Se puede observar que el tiempo de respuesta (response time) se mantiene constante para los primeros 5 minutos, debido a que no se podía conectar con el servidor y se alcanzaba el tiempo de timeout establecido en 2373 ms. Luego, se puede ver cómo el valor disminuye a medida que se logran realizar las solicitudes exitosas.

Implementación del sistema GRIN Global Community Edition para la gestión de los bancos de germoplasmas de la Red de Recursos Fitogenéticos del INTA
Proyecto Final de Grado - Escalante, Santiago

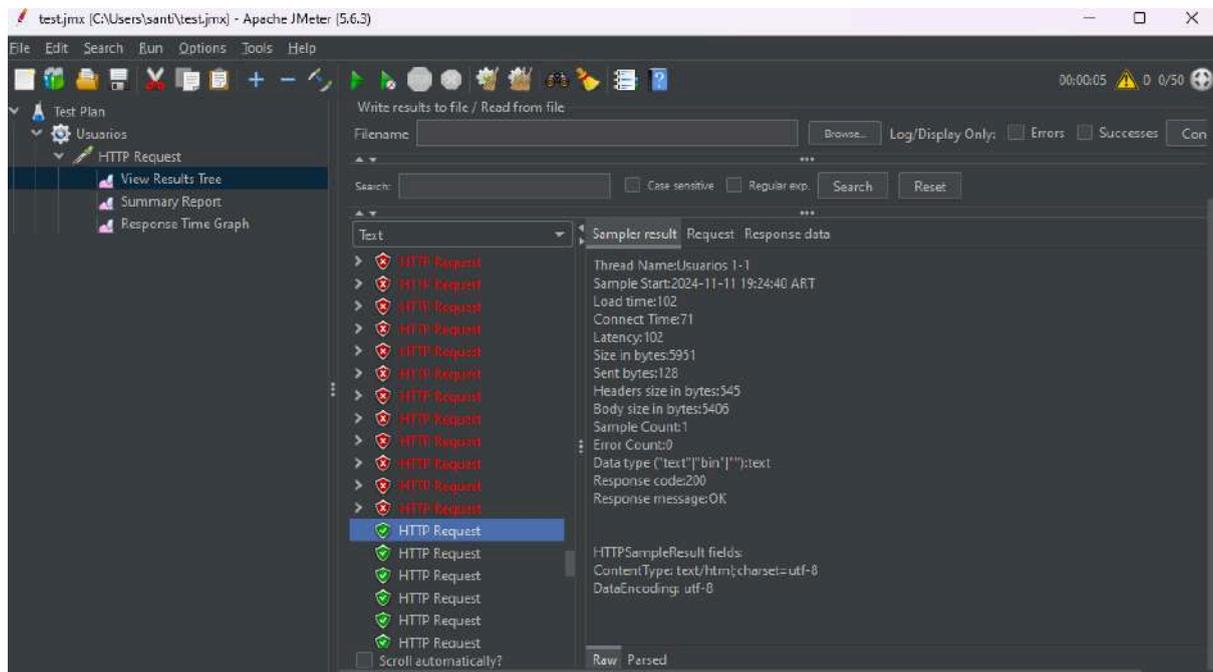


Figura 6. Muestra de respuesta OK (Código de respuesta 200).

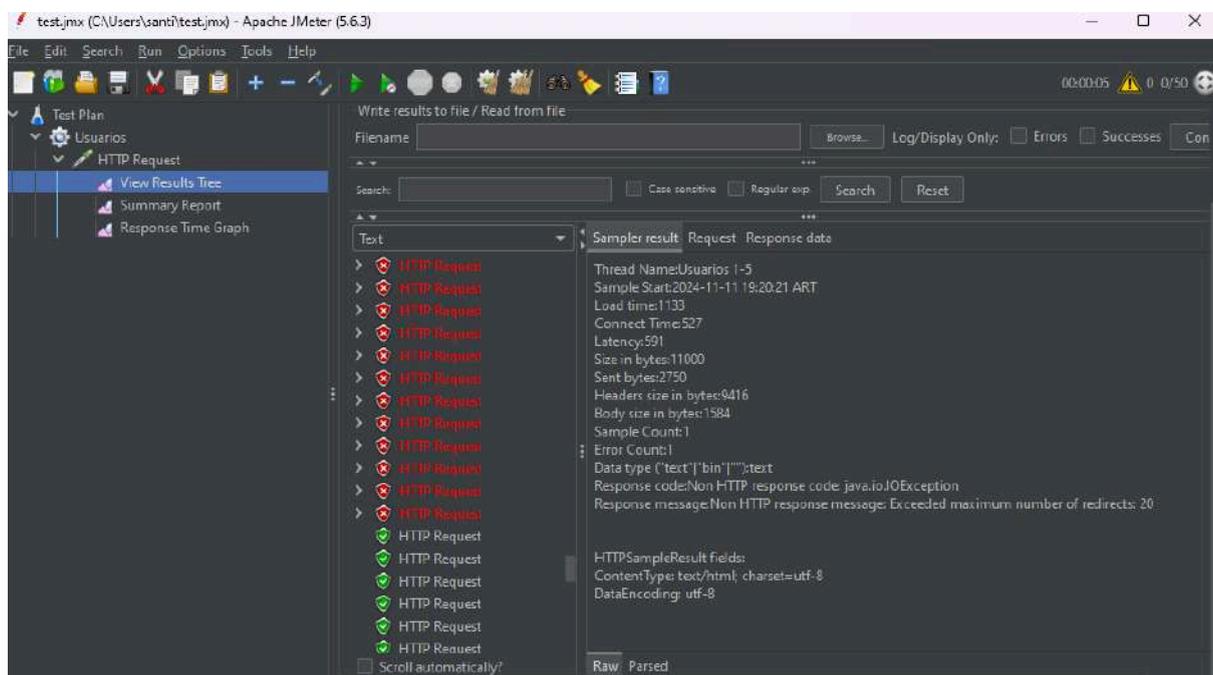


Figura 7. Muestra de respuesta con error por límite de reintentos.

Azure DevOps

Azure DevOps es un conjunto de herramientas integradas proporcionadas por la nube de Microsoft Azure que permiten gestionar todo el ciclo de vida de un proyecto de *software*. Es ideal para que los equipos puedan planificar, desarrollar, probar, controlar y entregar *software* de manera eficiente y de forma centralizada y colaborativa.

Principales características de Azure DevOps

Azure DevOps está conformado por una serie de servicios, cada uno con un enfoque diferente sobre el desarrollo de *software*:

- **Azure Boards:** Es una herramienta ágil de planificación y seguimiento de tareas. Cuenta con tableros Kanban, diagramas de Gantt, herramientas de scrum y reportes avanzados. Es ideal para la gestión de los elementos de trabajo, historias de usuario, *bugs* y *sprints*.
- **Azure Repos:** Permite la creación de repositorios de control de versiones de tipo Git y Team Foundation Version Control (TFVC). Permite aplicar *pull requests*, protección y estrategia de ramas y revisiones de control. Además, es posible la integración con herramientas externas de análisis y calidad de código, como SonarQube.
- **Azure Pipelines:** Permite la automatización de tareas relacionadas con la creación (pruebas y compilación, por ejemplo) y despliegue de aplicaciones

mediante *pipelines*. Una herramienta de integración continua y despliegue continuo(CI/CD) que soporta cualquier lenguaje, plataforma nube. Es compatible con herramientas como Docker, Kubernetes y Terraform. Además, ofrece integración con otras plataformas de repositorios de código como GitHub o BitBucket.

- **Azure Test Plans:** Herramienta para planificar y realizar pruebas manuales, exploratorias y automatizadas. Permite realizar el seguimiento detallado de bugs y generar reportes.
- **Azure Artifacts:** Es un gestor de paquetes para Maven, NPM, NuGet, entre otros. Facilita compartir y consumir paquetes para múltiples proyectos. Ideal para mantener dependencias y bibliotecas reutilizables.

Ventajas de Azure DevOps

Las ventajas de implementar Azure DevOps en los proyectos de una empresa u organización pueden ser los siguientes:

- **Plataforma completa:** Integra todas las fases del ciclo de vida en el desarrollo de *software*.
- **Flexibilidad:** Funciona con cualquier lenguaje, tecnología, o plataforma(Windows, Linux, MacOS).

- **Escalabilidad:** Adecuado tanto para equipos grandes como chicos.
- **Integración con herramientas externas:** permite el uso de herramientas como GitHub, Jenkins(CI/CD), Kubernetes, entre otras.
- **Opciones de despliegue:** Es posible utilizarlo en la nube o en su versión *on-premise*(local) con Azure DevOps Server.

Seguridad

Entre las medidas de seguridad del sistema, se destaca que el tráfico de datos entre los componentes se establece mediante SSL, lo que garantiza que la información se transmita de forma segura, evitando ser interceptada y descifrada. Además, se implementan dos capas adicionales de seguridad: una es la VPN, que permite el acceso de forma remota a la red interna mediante SSL, creando un túnel seguro en Internet. Para acceder a la red interna, primero hay que pasar el firewall implementado. En segundo lugar, para acceder al servidor del INTA, es necesario atravesar un proxy inverso que, en base a la petición y la URL, redirige el tráfico hacia destino. Cabe destacar que los puertos están cerrados hacia el exterior, por lo que es necesario solicitar la exposición solo de los puertos requeridos por la solución.

Arquitectura final con requerimientos de seguridad INTA

En el siguiente diagrama se observa la arquitectura, que fue evolucionando hasta dar en la arquitectura final del sistema. Como Cliente VPN se utiliza FortiClient y en la infraestructura del INTA se ha implementado un FortiGate. FortiGate es un

dispositivo que cumple la función de firewall e impide el acceso no autorizado. Además del firewall, se encuentra un proxy inverso que redirige el tráfico a los puertos correspondientes según la URL de las solicitudes.

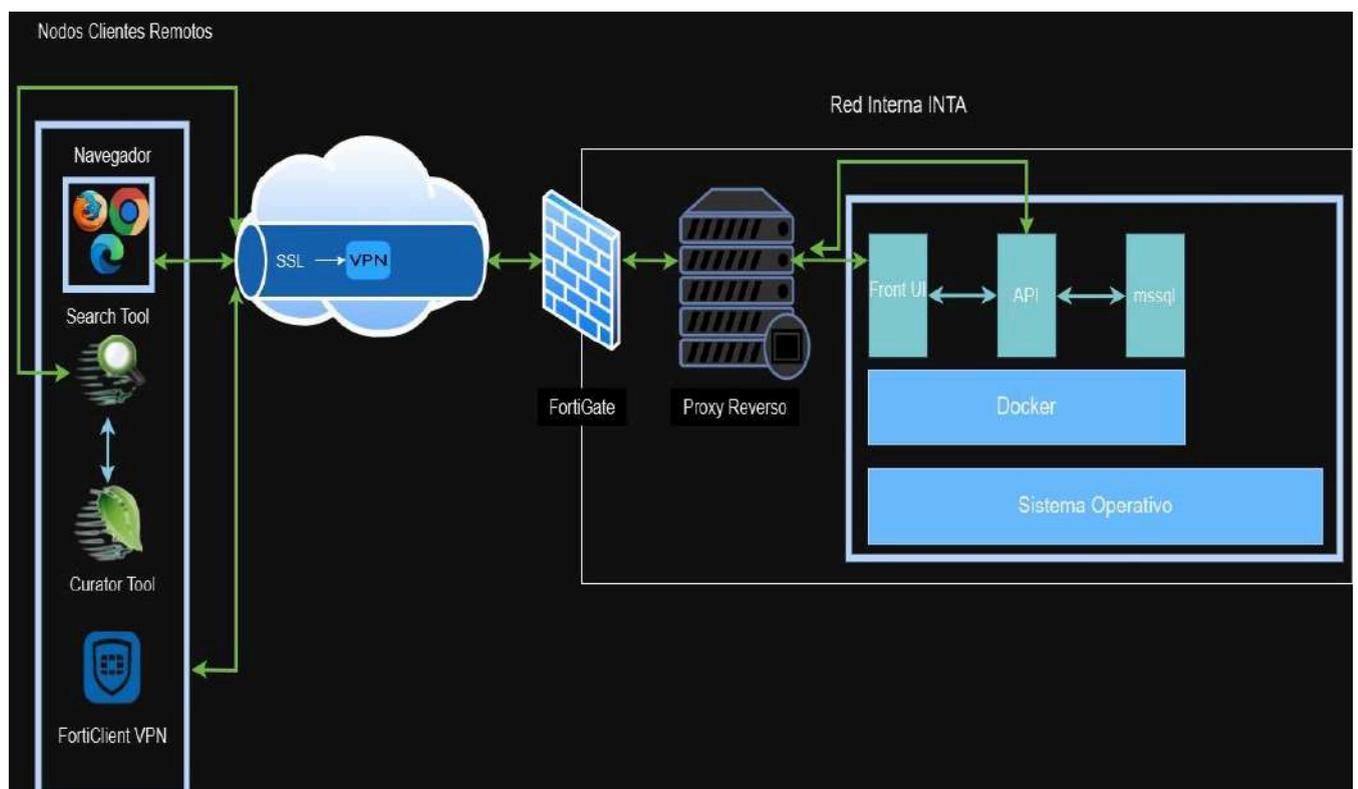


Figura 8. Diagrama de la solución con los requerimientos de seguridad impuesto por INTA.

Modelo de datos

El modelo de datos se encuentra definido en su totalidad y no requiere agregar nuevas entidades para modelar el sistema.

La siguiente figura muestra las distintas tablas que componen la base de datos. Las flechas indican la dependencia de cada tabla; por ejemplo, la tabla *Accession* depende de la tabla *Taxonomy Species*, que se considera como la "raíz". Otro aspecto a tener en cuenta es que observamos la dependencia de varias tablas con

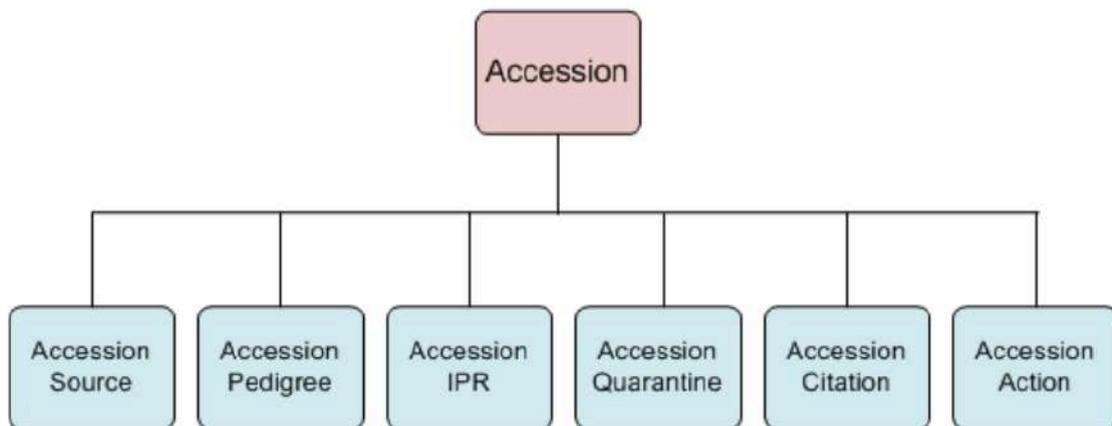


Figura 10. Sección del modelo de datos Parte Accesiones.

Resultado

Resultado obtenido

Como resultado se obtuvo la implementación del producto específico de GRIN Global especializado para bancos de germoplasma. La solución consta de un servidor con el *software* GRIN Global Community Edition y los clientes que tienen instalado la herramienta GRIN Global Curator Tool.

- GRIN Global Curator Tool(Cliente).

Es utilizado para la gestión de etiquetado e inventario por parte de los usuarios.

Implementación del sistema GRIN Global Community Edition para la gestión de los bancos de germoplasmas de la Red de Recursos Fitogenéticos del INTA
Proyecto Final de Grado - Escalante, Santiago

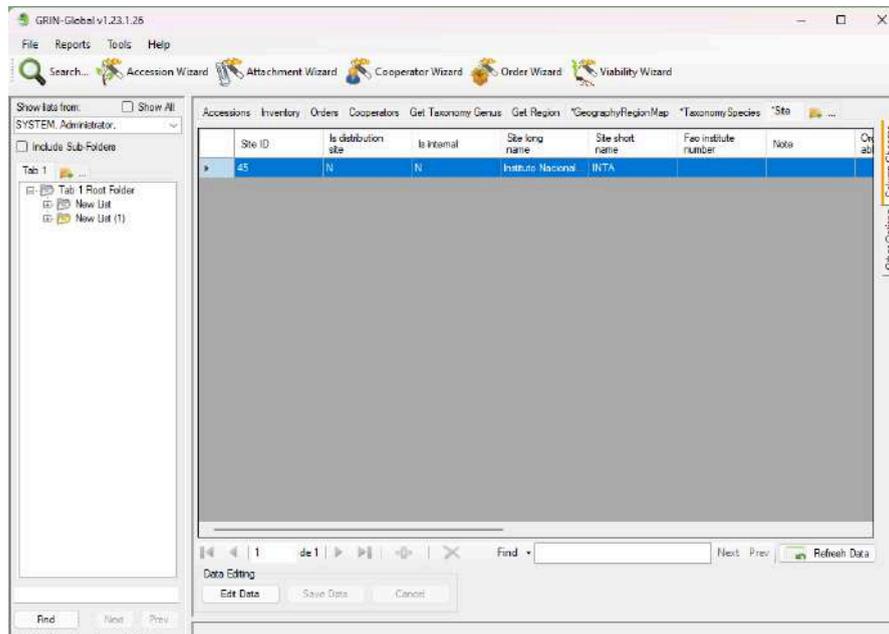


Figura 11. Vista de datos de la tabla **Sites** desde Curator Tool.

- GRIN Global Community Edition (Servidor).

Contiene de manera centralizada la información del banco de germoplasma.

Cada instancia del Curator Tool interactúa con él mediante un modelo

cliente/servidor.

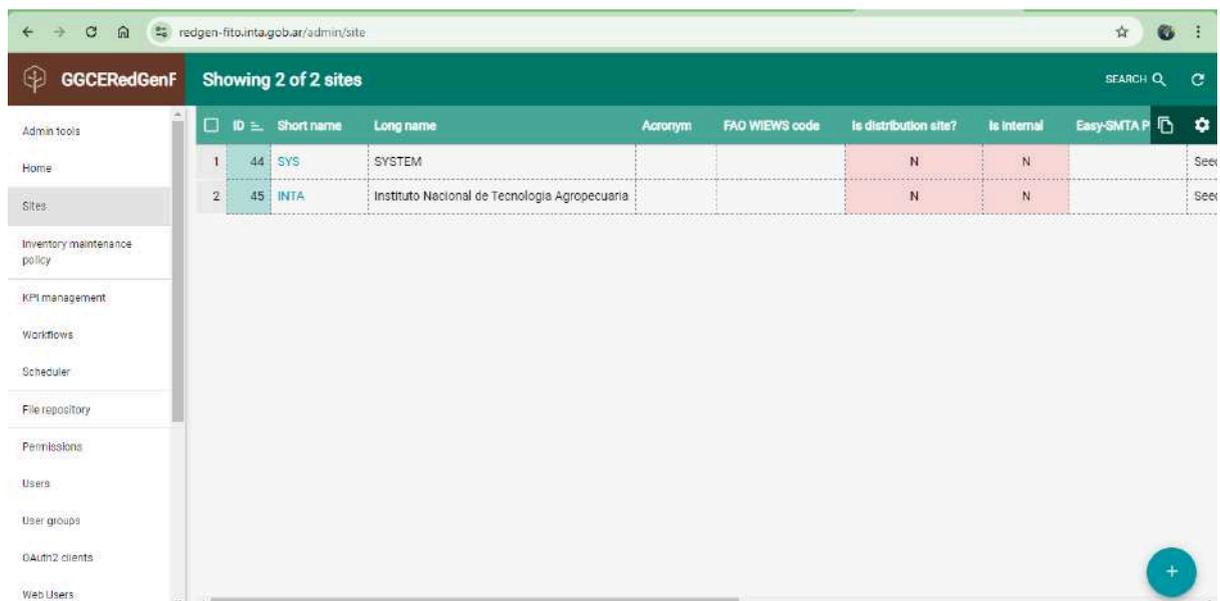


Figura 12. Visualizando los datos de la sección **Sites** desde el sitio web GGCE.

Trabajo futuro

Al definir el alcance del proyecto, no se contempló un soporte a largo plazo, ya que esto extendería considerablemente los plazos del trabajo. Tras la capacitación del personal, se atenderán las consultas que puedan surgir a medida que los usuarios hagan uso del sistema, durante un periodo de tiempo limitado.

No obstante, se han analizado posibles tareas adicionales que podrían agregar valor a la solución. Entre ellas se encuentran la automatización de los despliegues de la aplicación, en caso de que existan actualizaciones. Otra tarea podría ser aplicar monitoreo para obtener métricas y visualizar el estado de los distintos componentes del sistema (como el porcentaje de memoria utilizado, CPU, etc) para tener más control sobre la solución. También se considera la opción de integrar el sistema con herramientas de visualización y análisis de datos, como Power BI, para automatizar la recolección de datos que se utilizan con frecuencia en la actividad diaria de los investigadores.

Memoria del proyecto

Participación de la referente y parte interesada

Durante el transcurso del proyecto, se estableció un flujo constante de comunicación bidireccional con la referente funcional y su equipo. Esta interacción no se limitó solo a esperar el momento de la entrega de los documentos, sino que también se caracterizó por un genuino interés en apoyar el progreso y el éxito del proyecto. La referente funcional Ariana Digilio, junto con Raquel Defacio y Rosana Bubillo, han tenido una participación activa desde el inicio, siempre dispuestas para hacerse un tiempo, revisar los documentos y explicar términos propios del dominio, acompañados de breves definiciones.

A medida que avanzaba el proyecto, se organizaron reuniones para revisar el progreso y discutir posibles ajustes o mejoras. Estas sesiones se convirtieron en un espacio óptimo para el intercambio de ideas y la resolución de dudas o inquietudes que pudieran surgir en el equipo de implementación. Tanto la referente funcional como las partes interesadas se mostraron receptivas a las sugerencias del equipo y brindaron valiosos comentarios y recomendaciones que contribuyeron significativamente al éxito del proyecto.

Su colaboración se extendió a lo largo de todo el proceso, desde la recopilación de información hasta la puesta en marcha de la solución. En cada etapa, se contó con su respaldo y confianza, lo que permitió avanzar de manera eficiente y superar los obstáculos que pudieran surgir en el camino, optimizando los tiempos de desarrollo. Su compromiso y dedicación se reflejaron en la calidad del trabajo realizado y en los resultados obtenidos.

Cumplimiento de objetivos

Objetivo principal

Implementar el *software* GRIN Global Community Edition en la Red de Recursos Fitogenéticos perteneciente al INTA. Mediante la infraestructura requerida y utilizando la herramienta de contenerización, se pudo cumplir con el objetivo. El acceso a los datos no es más un problema para los integrantes de la red de recursos fitogenéticos, ya que ahora pueden acceder al servidor donde se encuentran los datos de forma centralizada. Además, se destaca que toda esa información estará disponible para los 21 bancos de germoplasma distribuidos en todo el país.

Objetivos específicos

Diseñar la arquitectura de la solución en los servidores del INTA y en los nodos clientes. Se determinó, mediante las reuniones con el representante del *software* los y la lectura de la documentación oficial de cada producto, los componentes necesarios para lograr definir la arquitectura del sistema en conjunto con los requerimientos de seguridad que corresponden al INTA y su infraestructura.

Implementar el *software* GRIN Global Community Edition. Se configuró y realizó el despliegue correspondiente del *software* en el servidor provisto. Por otro lado, se instaló el *software* cliente junto con sus dependencias en todos los nodos definidos. Además, se realizaron pruebas para determinar que la solución cumple con el comportamiento esperado.

Capacitar al personal que hará uso del sistema. Se llevaron a cabo una serie de reuniones con el personal para presentar el *software*, hacer una demostración de las funcionalidades y entregar la documentación necesaria para poder operar la herramienta. Finalmente, se logró que el equipo capacitado pueda utilizar el sistema de manera eficaz.

Planificación original vs ejecución

Inicialmente, el cronograma se planificó bajo un esquema de modelo en cascada, considerando 4 etapas con sus respectivas tareas por componente. Se estimó un total de 248 horas distribuidas en 18 semanas.

Aunque en principio se supuso un modelo en cascada, durante el desarrollo del proyecto no siempre se siguió este enfoque. Mientras que las etapas propias de la naturaleza del proyecto se ejecutaron de manera secuencial, algunas etapas, como la documentación final y la bitácora de trabajo, se llevaron a cabo en paralelo.

Al optar por una planificación orientada por componentes de *software*, el proyecto pudo ser gestionado de manera eficaz, permitiendo la entrega temprana de un producto de valor que los usuarios pudieron hacer uso antes de la finalización del proyecto.

Tras finalizar con la fase de instalación del servidor, los usuarios pudieron comenzar a utilizar el sitio web y, posteriormente, la instalación de los clientes añadió mayor funcionalidad a la solución. Esto evidencia que, además de aplicar una metodología en cascada, se priorizó la entrega temprana de *software*, con funcionalidades que

se incrementaron progresivamente a medida que avanzaba el proyecto. Una combinación efectiva de ambos enfoques.

La etapa de instalación de los nodos clientes se decidió postergar hasta contar con mayor experiencia en el campo y un mayor entendimiento del funcionamiento del sistema, con el objetivo de alcanzar la madurez necesaria para poder responder consultas y realizar demostraciones.

Otro aspecto importante a destacar es que, en la estimación inicial para la entrega del protocolo, no se incluyeron como etapas formales ni la documentación final ni la bitácora de trabajo, dejándolas fuera de los tiempos del proyecto. Este ha sido un error que servirá como enseñanza para los próximos proyectos.

En la siguiente figura se muestra la comparativa entre lo estimado y lo que realmente sucedió. Los valores dentro de las celdas representan la cantidad total de horas que se trabajó en ese periodo de tiempo. Es importante aclarar que, durante el receso invernal, periodo en el cual permanece cerrada la Facultad, no se ha avanzado con las tareas. Los motivos son los siguientes: primero, para tomarse vacaciones por parte del alumno; y segundo, no trabajar sin la validación o supervisión de los directores del proyecto.

Implementación del sistema GRIN Global Community Edition para la gestión de los bancos de germoplasmas de la Red de Recursos Fitogenéticos del INTA

Proyecto Final de Grado - Escalante, Santiago

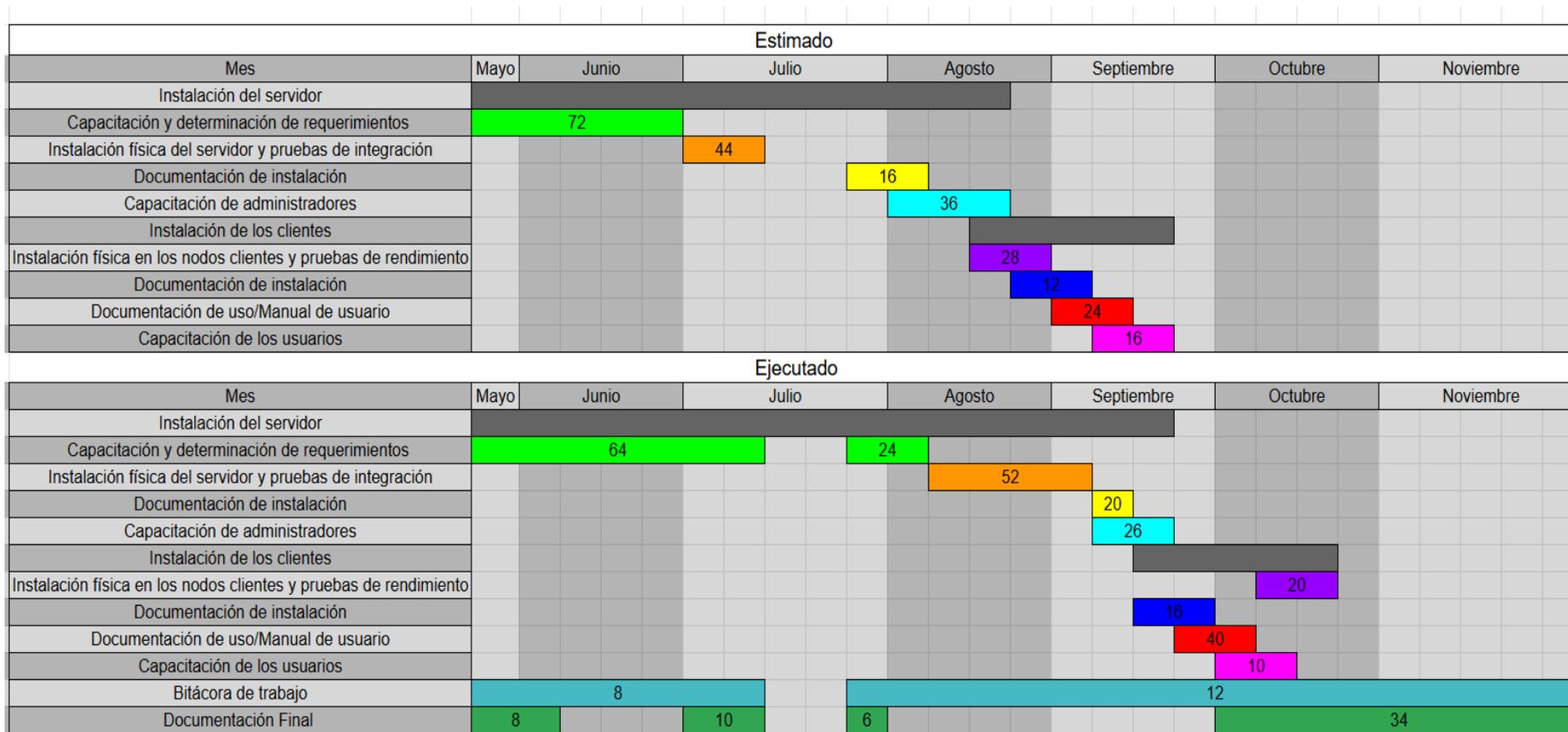


Figura 14. Planificación estimada vs Ejecución.

Análisis FODA

Al inicio del proyecto se realizó el análisis correspondiente, en el que se identificaron los puntos positivos y negativos para abordarlo. Una vez finalizado, se determinó lo siguiente:

Entre las debilidades enunciadas se notaron los siguientes cambios:

- “Poco conocimiento sobre el campo de interés”. El estudio de los bancos de germoplasma y el campo de la genética se llevó a cabo como parte de la etapa de investigación del proyecto y se complementó con consultas a la referente funcional. Esto ya no representa una debilidad para el equipo.

Entre las amenazas previamente definidas se observó lo siguiente:

- “El demandante se decida por otra oferta”. El demandante mantenía contacto constante con el representante de la empresa por lo que era poco probable que esto ocurriera. Otros factores podrían haber afectado la selección del *software*, como los requerimientos de seguridad, el uso de *TLS*, por ejemplo. Tras investigar la solución, se descartó toda posibilidad de que esto ocurriera.
- “La solución la lleve a cabo otro equipo de profesionales”. Inicialmente, existía la posibilidad de que, si el equipo no lo hacía, otro tendría que hacerlo. Se consideraba una amenaza la competencia de los profesionales con los que cuenta el INTA, quienes tienen más experiencia. Sin embargo, debido al compromiso y la confianza del equipo para avanzar con las tareas, esta amenaza fue desapareciendo.
- “La burocracia administrativa por parte del demandante, procesos lentos que dificultan la agilidad operativa”. En ocasiones, se sabía qué hacer, pero los

obstáculos no permitían actuar. Un ejemplo de esto fue la parte de las instalaciones de los clientes en los nodos de trabajo, que requirieron del administrador para poder tener privilegios para instalarlo. Para solicitar la presencia del administrador, los investigadores tuvieron que levantar un ticket, el cual tuvo un tiempo de respuesta promedio de una semana. Esto generó la aparición de esta nueva amenaza.

Durante el desarrollo del proyecto, se identificó una nueva fortaleza:

- “La participación de Adolfo Spinelli fue clave para lograr cumplir con los objetivos”. Al haber trabajado anteriormente con el INTA, conocía cómo trabajan y la infraestructura con la que contaban. Además, tenía experiencia en implementaciones de este tipo, por lo que sabía qué requerimientos y tareas eran fundamentales. Su aporte fue muy importante.

Riesgos y Solventación

Durante el desarrollo del proyecto se materializaron los riesgos R02 y R03 definidos en la sección de [análisis de riesgos](#). Para ambos, se actuó siguiendo lo establecido en los [planes de contingencia](#).

El riesgo R02 podría haber tenido un alto impacto en la planificación y en los tiempos de trabajo. Sin embargo, el rol activo del equipo y de la referente funcional, que en ciertos momentos se encontraban con pocos tiempos debido a sus tareas diarias, ha permitido que la comunicación se lleve por otras vías más rápidas aunque a veces no tan efectivas. Esto evitó un estancamiento por estar a la espera de órdenes o retroalimentación.

En cuanto al riesgo R03, es importante mencionar que el integrante del equipo estuvo de viaje por una semana. No obstante, esta situación, que fue definida en el plan de contingencia oportunamente planteado, no tuvo impacto en los tiempos del proyecto.

Por otro lado, los demás riesgos planteados no tuvieron lugar durante el transcurso del proyecto y tampoco surgieron riesgos no contemplados inicialmente.

Análisis de etapas

Instalación del Servidor

Para la instalación del servidor, se incluyeron tareas de investigación y capacitación por parte del estudiante, con el fin de comprender la arquitectura, los requerimientos y el despliegue del servidor. Asimismo, se llevaron a cabo diversas reuniones con el área de ciberseguridad del INTA, la referente funcional, el representante del *software* y otros interesados. Posteriormente, se realizó el despliegue del servidor, seguido de las pruebas de integración con la base de datos y el cliente, la documentación del proceso de instalación y administración y, finalmente, la capacitación del administrador del INTA.

Capacitación y determinación de requerimientos

Durante esta etapa, se realizaron tareas de estudio de las tecnologías requeridas, como la herramienta Docker para la contenerización de aplicaciones, con el objetivo de comprender su funcionamiento y uso. Además, se dedicó tiempo a investigar la instalación del servidor de la solución. También, se llevaron a cabo reuniones con el equipo de ciberseguridad del INTA, el representante del producto y la referente funcional, con el fin de presentar el proyecto, aclarar dudas sobre la herramienta y definir los requerimientos, especialmente los de seguridad, que debían cumplirse de manera estricta de acuerdo con las políticas del INTA.

El representante del *software* organizó una capacitación para explicar la arquitectura de la solución y enseñar el uso del cliente. En paralelo, el representante del área de ciberseguridad solicitó que se enviaran los requerimientos de *hardware*, el protocolo de comunicación entre el cliente y el servidor y la arquitectura previamente mencionada. Una vez recopilada la información solicitada, se envió al INTA y, tras unas semanas, se recibieron las credenciales para acceder a la infraestructura donde se desplegará el servidor.

Con las credenciales en mano, se probaron los accesos. Para conectarse al servidor, fue necesario descargar un cliente de *VPN* que establece una conexión segura con la red del INTA. La conexión se habilita al iniciar sesión con un usuario registrado en el Directorio Activo de la institución. Una vez establecido el túnel seguro, es posible conectarse al servidor mediante el protocolo *SSH*, utilizando su IP local en el puerto 22.

Por último, se informó a INTA los nombres de dominio requeridos para la solución, uno para la *API* y otro para el sitio web con sus respectivos puertos a los que redirigir el tráfico.

Instalación física del servidor y pruebas de integración

Para continuar, se instalaron las dependencias, como Docker, en el servidor provisto. Posteriormente, se descargaron las imágenes del *software* y una imagen de un servidor de base de datos para comenzar con el despliegue. El proceso requiere dos contenedores: uno para la *API* y otro para la página web a la que pueden acceder los usuarios demandantes.

Se informó al INTA sobre los nombres de dominio requeridos para la solución, uno asociado a la *API* y otro al sitio web, junto con los puertos a los que se debía redirigir el tráfico. Una vez adaptados los archivos de configuración a la infraestructura existente, se procedió con el despliegue.

Sin embargo, hubo problemas debido a que el INTA cuenta con una implementación de proxy inverso, mientras que la solución también incluía su propio proxy inverso. Al desplegarlo, no era posible acceder a ninguna de las dos URLs. Esta situación fue reportada al equipo de ciberseguridad del INTA, quienes identificaron que el proxy inverso adicional podría ser la causa del conflicto.

Para resolver el problema, se decidió eliminar la implementación del proxy inverso incluida en la solución y nuevamente intentar desplegar el servidor. Esta ejecución resultó exitosa y el servidor quedó accesible para los usuarios.

Finalmente, se realizaron pruebas de integración con la base de datos desplegada. Para ello, se hizo una carga de datos de prueba desde la página web y se verificó que los datos ingresados luego eran persistidos, lo que confirmó que la conexión entre ambos componentes era correcta.

Documentación de instalación

Se elaboró un entregable que contenía toda la información necesaria para instalar el servidor. En este se incluyeron los enlaces para descargar las dependencias y el *software*, junto con la configuración personalizada, los comandos requeridos y recomendaciones de administración y seguridad.

Capacitación de administradores

Se llevó a cabo una reunión con el equipo del área de ciberseguridad del INTA, en la cual se entregaron las credenciales del usuario administrador, el documento de instalación y otro documento redactado para la administración del sistema. Durante la reunión, también se presentó la configuración de los archivos, junto con una demostración del despliegue, el acceso al sitio web y las principales funcionalidades de administración propias de la herramienta.

Por su parte, los representantes del INTA plantearon consultas sobre el crecimiento de la base de datos, la cual deberá ser monitoreada para realizar un seguimiento y resolver posibles problemas de rendimiento a futuro.

Además, se solicitó que todo el material relacionado con el proyecto se subiera a la nube de la organización en Azure, utilizando el conjunto de herramientas que brinda Azure DevOps para su gestión.

En Azure DevOps se realizaron las siguientes tareas a pedido de las personas del INTA:

- Los archivos requeridos para el despliegue se subieron a los repositorios de INTA en Azure Repos. Cada proyecto administrado por el INTA tiene su propio repositorio, donde se almacena el código fuente.
- Se generó una página de la *wiki* en la sección de Azure Overview, que contendrá toda la información relevante sobre el proyecto, facilitando el acceso a todos los documentos generados como material adicional para que el administrador designado pueda comprender el sistema.

Asimismo, las imágenes de Docker utilizadas fueron subidas al registro de contenedores propio de la organización en la nube, lo que permite un control centralizado y seguro de las imágenes empleadas dentro la institución.

Instalación de los clientes

Durante el desarrollo de esta etapa, se realizaron reuniones con las partes interesadas para poder acceder de manera remota a los puestos de trabajo para la instalación. Aunque se estimó que el número de personas que hará uso de la herramienta es de aproximadamente 30, solo se trabajó directamente con 3 integrantes del equipo para llevar a cabo las instalaciones correspondientes, ya que resulta inviable que una sola persona asumiera la instalación del cliente para todos los usuarios. En su lugar, se delegó la responsabilidad, proporcionando a cada integrante el soporte necesario para realizar su propia instalación.

Para facilitar este proceso, se redactó la documentación de instalación, junto con los manuales de usuario para el uso del sitio web y del cliente Curator Tool. Finalmente, se llevó a cabo una capacitación con los miembros mencionados, en la que se entregó toda la documentación y se realizaron demostraciones prácticas para comprender los flujos y las funcionalidades que ofrece la solución. Luego, se permitió que los investigadores iniciaran con la carga de datos al sistema, brindando asistencia durante el proceso hasta que lograran adaptarse a la herramienta.

Además, se realizaron las pruebas de rendimiento para validar que el sistema fuera capaz de responder de forma correcta a escenarios en los que se atraviesa por picos de demanda.

Instalación física en los nodos clientes y pruebas de rendimiento

La instalación de los clientes tuvo ciertos obstáculos. En primer lugar, los usuarios utilizados por los investigadores en los puestos de trabajo no contaban con privilegios de administrador. Quien tiene permisos para poder descargar *software* en las máquinas es el administrador en INTA, por lo que fue necesario realizar una solicitud por ticket, que demoró una semana hasta obtener una respuesta.

Además, cada computadora no contaba con el cliente VPN requerido para acceder al servidor donde se alojan el sitio web y la base de datos, por lo que requirió configurarlo en cada uno de los 3 puestos de trabajo. Finalmente, se instaló el cliente Curator Tool y se configuró la conexión con el servidor, completando así las tareas de instalación. Se generaron las credenciales para cada uno de los usuarios y se verificaron los accesos, asegurando su correcto funcionamiento.

En cuanto a las pruebas de rendimiento, se utilizó la herramienta JMeter para simular solicitudes concurrentes de múltiples usuarios para verificar si el sistema respondía correctamente. Los resultados fueron exitosos, confirmando que la solución soportará el acceso concurrente y distribuido como se esperaba.

Documentación de instalación

Como se mencionó anteriormente, la posibilidad de instalar el cliente en todos los puestos de trabajo se vio limitada debido a la cantidad de personas interesadas y la disponibilidad de cada una. Por ello, se optó porque las demás partes hicieran su propia instalación, siguiendo los pasos establecidos en el documento redactado y entregado a la referente funcional, quien se encargará de distribuirlo para que todos

los interesados puedan acceder a la información. Además, se incluyeron ejemplos de uso de la herramienta para poder empezar a utilizarla.

Documentación de uso/Manual de usuario

Para aprovechar todas las funcionalidades que provee el sistema y sus componentes, se escribieron dos manuales en los que se detalla el uso del cliente y la página web. Ambos servirán como guía para que los investigadores logren adaptarse a las herramientas y puedan sacarle el máximo provecho. Además, permitirá ampliar el conocimiento que tienen sobre el sistema.

Capacitación de los usuarios

Finalmente, se organizaron encuentros con la referente funcional y dos investigadores, quienes conforman la parte interesada. Durante estos encuentros, se llevaron a cabo demostraciones del uso de las herramientas y se mostró cómo está estructurado el modelo de datos y la arquitectura del sistema para entender los flujos y la relación entre cada componente. Además, se hizo entrega de los manuales para que puedan utilizar como material adicional para ayudarles a entender y hacer uso del sistema, así como para distribuirlo con el resto de los investigadores. Se ofreció un periodo de soporte para dar asistencia hasta que logren adaptarse al nuevo sistema y puedan llevar a cabo sus tareas de manera exitosa.

Documentación final

En esta etapa, se consideraron los tiempos para la redacción del protocolo del proyecto, del informe final y la revisión de la documentación generada durante las

distintas fases del proyecto, la cual será entregada en forma de anexo para la cátedra del Trabajo Final.

Junto con la bitácora de trabajo, estas son las dos etapas que no se tuvieron en cuenta en la estimación inicial, lo cual se considera un error, ya que fueron piezas claves para lograr comunicar de manera eficaz el proyecto realizado.

La escritura del informe fue lo que más tiempo requirió. Cabe destacar que la bitácora ha hecho que la redacción del informe final sea más rápida y precisa debido a la información relevante del proyecto que contenía.

Además, se incluyeron los tiempos dedicados a correcciones solicitadas por los directores del proyecto sobre la redacción, el enfoque utilizado y qué secciones podrían agregarse, quitarse o modificarse para la entrega final.

Bitácora de trabajo

Si bien se mencionó anteriormente que esta etapa no fue incluida en el diagrama de Gantt presentado en el protocolo de trabajo final, la bitácora de trabajo se considera una herramienta fundamental para documentar lo realizado y dejar evidencia de ello.

El objetivo de la bitácora es recopilar toda la información y los acontecimientos relevantes necesarios para redactar el informe, mejorando así la gestión del tiempo y la eficiencia. Además, contribuye a garantizar que el contenido del informe sea fiel a lo ocurrido durante las distintas etapas del proyecto.

La bitácora se inició el primer día del proyecto, más precisamente cuando se fueron listando las preguntas para la primera reunión con la referente y se finalizó con lo

ocurrido durante las tareas de las pruebas de rendimiento del sistema. En total, fueron 12 horas las que llevó su redacción.

Fue de gran utilidad para determinar el inicio y fin de cada una de las etapas, identificando desviaciones debido a problemas que fueron ocurriendo en el transcurso del proyecto y que se detallaron en el registro, junto con las decisiones tomadas.

Sin embargo, la bitácora podría haberse mantenido de una manera más ordenada. La información fue ingresada sin un formato adecuado, lo que resultó algo desprolijo. También, faltaron algunos registros sobre los tiempos de desarrollo de las tareas, lo cual será tenido en cuenta a futuro para lograr una mejor gestión.

Se adjunta como [anexo](#) el contenido de la bitácora.

Aporte del proyecto

Utilización de nuevas tecnologías

Según la definición del perfil del ingeniero argentino: «El graduado de ingeniería debe poseer una adecuada formación científica, técnica y profesional que lo habilite para ejercer, aprender, desarrollar y emprender nuevas tecnologías, con actitud ética, crítica y creativa para la identificación y resolución de problemas en forma sistémica, considerando aspectos políticos, económicos, sociales, ambientales y culturales desde una perspectiva global, tomando en cuenta las necesidades de la sociedad.»

El proyecto final es una experiencia donde se ofrece una valiosa oportunidad de aprendizaje y desarrollo para los estudiantes. En este caso, los conocimientos aprendidos proporcionan una sólida base para comprender y trabajar con herramientas enfocadas al desarrollo, despliegue de aplicaciones y pruebas de rendimiento. Si bien estas tecnologías no estaban definidas en principio, fueron seleccionadas a medida que se investigaba la solución y se definían las tareas a realizar.

En primer lugar, Docker es una herramienta poderosa que permite ejecutar aplicaciones modernas en diferentes entornos sin depender de una arquitectura en particular y haciendo un uso más eficiente de los recursos. Es ampliamente utilizado en aplicaciones basadas en microservicios que constituyen lo que hoy conocemos como el mundo *cloud* y funda las bases para implementar tecnologías de orquestación de contenedores como *Kubernetes*.

En segundo lugar, JMeter es una herramienta basada en Java utilizada principalmente para realizar pruebas de rendimiento sobre diferentes sistemas, validando su comportamiento en distintos escenarios. Es una herramienta que se desconocía y que investigando acerca de cómo realizar pruebas del sistema apareció como una posible solución. Ha sido una herramienta muy intuitiva y sencilla de aprender que será tenida en cuenta para futuros proyectos.

En tercer lugar, Azure DevOps es una herramienta muy interesante para la gestión de proyectos de Software, que ofrece todo lo necesario para que los equipos puedan trabajar de manera centralizada y colaborativa. Ofrece funcionalidades que permiten abarcar todas las tareas necesarias para el correcto desarrollo de software, con el objetivo de hacer que el proceso cada vez sea más eficiente.

Por ejemplo, permite la automatización de tareas como pruebas, compilación y despliegue a través de Azure Pipelines. Además, facilita la integración con otros servicios nativos de la nube, como Azure Container Registry, e incluso con servicios de otros proveedores, como Amazon Web Services o Google Cloud Platform. También admite la integración con las tecnologías mencionadas anteriormente, lo que la convierte en una herramienta muy poderosa.

Finalmente, el aporte de adquirir la experiencia necesaria para utilizar estas herramientas puede ser de gran ayuda en el ámbito laboral, ya que hoy en día son empleadas por muchos profesionales en el desarrollo y en gestión de software e infraestructura.

Aprendizaje

Además de las tecnologías utilizadas, que necesitaron de conocimientos previos, adquiridos a lo largo del proceso de formación, este proyecto deja una enseñanza muy importante sobre ciertos aspectos dentro la informática, como es el rol del Ingeniero DevOps. Este rol se caracteriza por su versatilidad, ya que combina el desarrollo de *software* con su despliegue, mantenimiento y gestión.

Aunque en este caso no hubo desarrollo de un software en sí, comprender cómo funcionan, los puertos utilizados y los errores comunes de las aplicaciones desarrolladas en diferentes lenguajes de programación es fundamental para esta actividad y hace más dinámico el proceso de entrega del software. En muchas ocasiones, lograr que una aplicación funcione correctamente en un entorno de producción real resulta más complicado de lo que parece. Por ello, es importante destacar esta actividad.

Antes de comenzar con el desarrollo de este proyecto, se asistió a la cursada de la materia Trabajo Final, en la que se presentaron diferentes definiciones sobre el perfil del ingeniero y las capacidades necesarias para obtener el grado correspondiente. A continuación, se enumeran las Competencias Específicas del Plan de Estudios, y se indicará si se han alcanzado a través de la realización del proyecto.

✓ Identificar, formular y resolver problemas de ingeniería informática.

En este caso, surgió la necesidad de informatizar y centralizar los datos de los bancos de Germoplasma. Este desafío fue resuelto con éxito mediante la implementación del software

✓ Concebir, diseñar y desarrollar proyectos de ingeniería informática (sistemas, componentes, productos o procesos).

Se diseñó una arquitectura de sistema basada en un modelo cliente-servidor, utilizando contenedores para una mayor eficiencia y escalabilidad.

✓ Gestionar, planificar, ejecutar y controlar proyectos de ingeniería informática (sistemas, componentes, productos o procesos).

Se gestionaron, planificaron, ejecutaron y controlaron las diferentes fases del proyecto. Este informe da testimonio del trabajo realizado y los logros alcanzados.

✓ Usar de manera eficaz las técnicas y herramientas de la ingeniería informática.

Se utilizaron nuevas herramientas de forma eficaz, tales como Docker, Azure DevOps, JMeter, todas ellas promueven técnicas para el desarrollo ágil.

✓ Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.

En este caso, como se mencionó anteriormente, el factor de innovación recae en que un banco de germoplasma de esta magnitud es algo nuevo a nivel local.

✓ Desempeñarse de manera efectiva en equipos de trabajo.

Si bien se contó con un solo estudiante, el equipo lo conformaron los codirectores con los cuales se logró un desempeño efectivo.

✓ Comunicarse con efectividad.

Se mantuvo una comunicación eficaz con todos los actores involucrados en el sistema, asegurando que las ideas y la información relevante fueran transmitidas de manera clara y comprensible.

✓ Tener capacidad para la toma de decisiones usando técnicas modernas de gestión.

Se tomaron decisiones clave relacionadas con la arquitectura y la planificación del proyecto. Además, se realizó un análisis FODA para gestionar los riesgos y desarrollar políticas para mitigarlos.

✓ Interpretar el sentido económico de las decisiones al adoptar alguna tecnología.

Las decisiones de tecnología fueron tomadas considerando el aspecto económico.

Se utilizaron herramientas *open source*, y al utilizar Docker para el despliegue mediante contenedores, se optimizó la infraestructura en cuanto a recursos, como núcleos del procesador, memoria RAM y almacenamiento.

✓ Ser consciente de la necesidad de una capacitación continua.

El proyecto fue planificado teniendo en cuenta la capacitación continua, tanto en el sistema a implementar como en las nuevas tecnologías que se utilizaron.

✓ Conducir y administrar recursos físicos y humanos en el marco legal y ambiental apropiado.

Se administraron los recursos físicos y humanos con los que se contaron en el proyecto de manera eficiente.

✓ Ser capaz de analizar la realidad informática internacional, nacional, regional y local y contribuir con soluciones informáticas a los problemas que surjan en dichos ámbitos.

El proyecto tiene una implementación de alcance nacional, específicamente para los bancos de germoplasma del Instituto Nacional de Tecnología Agropecuaria (INTA), contribuyendo al desarrollo del sector.

✓ Participar en actividades de investigación en el área informática para fomentar el desarrollo tecnológico de la región y el país.

Este proyecto contribuye al desarrollo tecnológico, informatizando y centralizando la información de los distintos bancos de germoplasma del país.

✓ Contribuir con la definición de políticas y estrategias en el área informática.

Se aportó en la definición de políticas de administración del sistema y en la creación de estrategias de despliegue basadas en contenedores para garantizar la escalabilidad y eficiencia del proyecto.

Resumen

En la siguiente figura se muestran una comparativa entre los tiempos estimados para cada etapa y lo ocurrido en la ejecución:

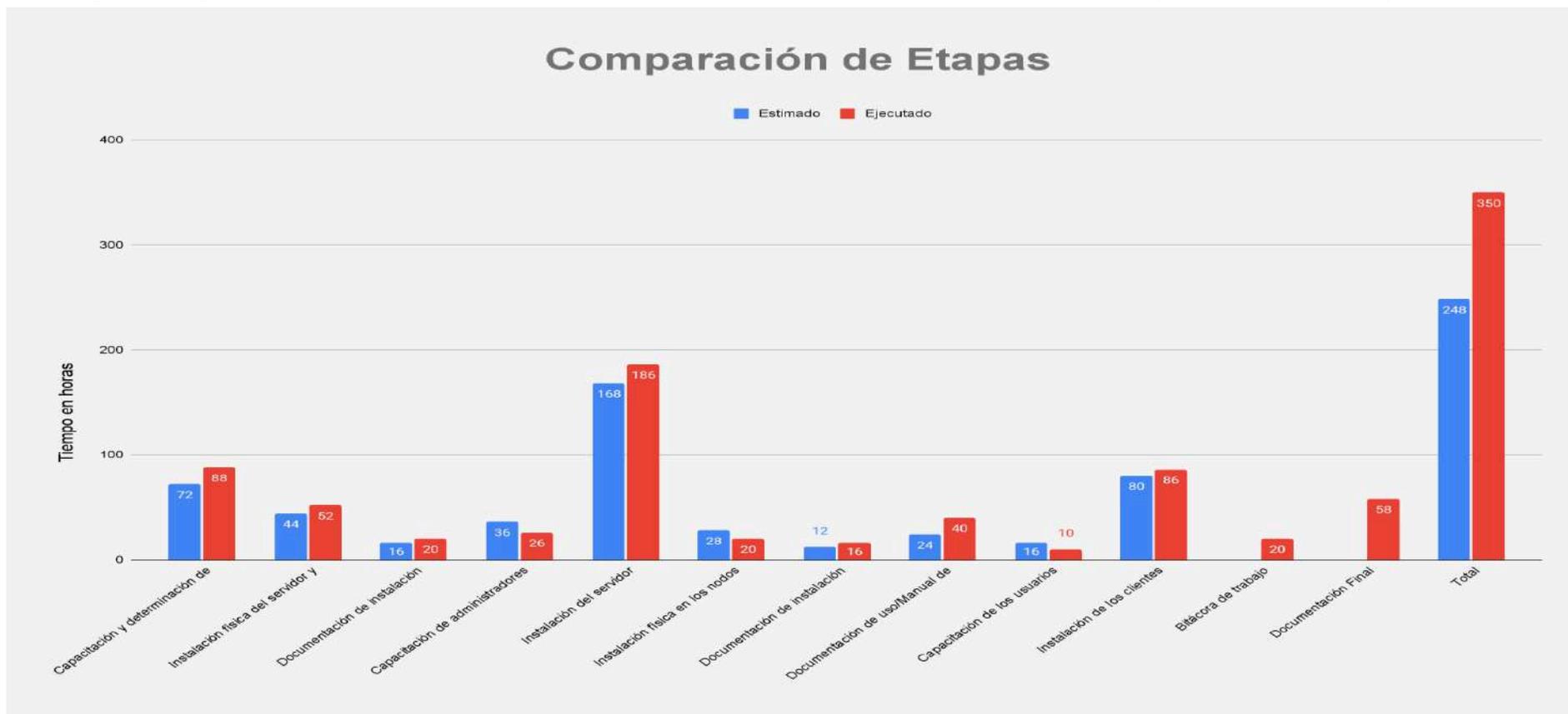


Figura 15. Tiempos estimados vs Tiempos de ejecución(En horas).

En total se registró un desvío de unas 24 horas sobre las etapas definidas inicialmente, lo que representa casi un 10 % más de lo esperado. La estimación se ve afectada por una tendencia a subestimar, aunque se observa también sobreestimación en algunas etapas. Esto último se ve reflejado debido a una cantidad menor de reuniones de las previstas y la falta de experiencia a la hora de estimar los tiempos de las tareas que se llevarán a cabo.

La omisión de estas etapas en la estimación inicial del proyecto, sumada a ciertos retrasos y problemas que ocurrieron a lo largo del proyecto, ha impactado en los tiempos ejecución de otras etapas, afectando la dedicación semanal esperada y ocasionando que el proyecto se extienda más allá de lo planificado.

Durante la primer etapa, se experimentó un retraso debido a la espera de que el INTA provisionara la infraestructura y las credenciales necesarias. En ese periodo, se continuó con la investigación y capacitación de las tecnologías a utilizar y el sistema a implementar. Horas invertidas que han sido necesarias para luego recuperar los tiempos perdidos.

Finalmente, al haber desvíos en todas las etapas, ya sea por subestimación o sobreestimación, esto no debe pasarse por alto. Ambas situaciones pueden ser perjudiciales en cualquier tipo de proyectos.

Los riesgos que se corren por **subestimar** son:

- Puede provocar retrasos en la entrega del proyecto y al incumplimiento de plazos, lo que afecta la satisfacción del demandante y la imagen del equipo. En estos

casos, pueden identificarse cuellos de botella en ciertas etapas, donde las tareas se realizan de manera más lenta debido a una asignación de recursos incorrecta.

- Los plazos ajustados pueden generar cierta presión, lo que podría afectar negativamente la calidad del trabajo.
- La subestimación puede llevar a costos adicionales inesperados, ya que pueden surgir problemas o tareas no previstas que requieren recursos adicionales para su realización.

Y por **sobreestimar** pueden ocurrir las siguientes situaciones:

- Es posible que se asignen más recursos de los necesarios (profesionales, *hardware*, *software*, etc), lo que podría llevar a un uso ineficiente de los recursos del proyecto. En consecuencia, los costos de proyecto pueden ser más altos, poniendo en riesgo su continuidad. Un análisis de viabilidad del proyecto podría categorizar como no rentable a un proyecto que, en realidad, sería viable si no se hubiera incurrido en la sobreestimación de tiempos y recursos.
- Si se sobreestiman los tiempos, en relación al punto anterior, pueden perderse oportunidades frente a competidores que estén capacitados a cumplir los plazos más ajustados.

Conclusiones

El objetivo principal del proyecto era implementar un software en la Red de Recursos Fitogenéticos perteneciente al INTA. Como así también diseñar la arquitectura de la solución en los servidores del INTA y en los nodos clientes, implementar el software GRIN Global Community Edition y capacitar al personal que hará uso del sistema.

Esto ha sido logrado gracias a la utilización de las herramientas adecuadas, como lo fue la herramienta de contenerización Docker, el constante apoyo por parte de los directores, la predisposición de la referente funcional y las demás partes interesadas, así como también la ayuda brindada del área de ciberseguridad y el representante del software.

Además, durante el desarrollo del proyecto, se logró poner en práctica los conceptos y habilidades adquiridas a lo largo de la carrera. Haciendo referencia particularmente a las asignaturas: “Redes de Computadoras”, “Software Libre”, “Gestión de la Seguridad Informática”, etc. Es importante mencionar la definición del perfil del ingeniero informático según la Universidad Nacional de Mar del Plata (UNMDP) donde cita lo siguiente: «Es una persona profesional con sólidos fundamentos teóricos y prácticos en el uso e implementación de las tecnologías relacionadas con la informática en aspectos como redes, sistemas distribuidos, arquitectura de procesadores y sistemas operativos utilizando como soporte estudios avanzados en algoritmos y sus aplicaciones. La formación debe permitir también la participación en actividades de investigación, desarrollo e innovación dentro de la disciplina»

Por otro lado, una de las tareas más difíciles de llevar a cabo fue la planificación y estimación de los tiempos. Si bien la cantidad total de horas dedicadas al proyecto se mantuvo dentro de un rango aceptable, la distribución de tiempos para cada una de las etapas, la cantidad de etapas contempladas y la dedicación semanal no fue la esperada, lo que generó que la duración del proyecto se extendiera varias semanas. La causa principal de estas desviaciones fue la falta de experiencia del integrante, debido a que no había participado de una planificación de esta magnitud anteriormente.

En conclusión, el resultado fue exitoso. Los objetivos planteados fueron cumplidos en su totalidad. Asimismo, ha quedado en evidencia la enorme importancia de la planificación y aplicación de metodologías pertinentes para la correcta ejecución de un proyecto de software. En lo personal, el proyecto representó una gran oportunidad para poner en práctica lo aprendido durante este periodo de formación. Fue una experiencia sumamente satisfactoria, que no solo permitió adquirir conocimientos y herramientas útiles para futuros proyectos, sino que también brindó al estudiante una valiosa experiencia profesional.

Apéndice

Glosario

Accession(*Accesión*). Es una muestra particular de material genético que se registra en los bancos de germoplasma y que forma parte de una colección.

API. Del inglés, Application Programming Interface, es una herramienta que permite a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos. La interfaz puede considerarse como un contrato de servicio entre dos aplicaciones que define cómo se comunican entre sí mediante solicitudes y respuestas.

Banco de germoplasma. Espacio que conserva la diversidad genética de especies agrícolas, forestales, pecuarias y microbianas. Su objetivo es preservar el mayor número de plantas, semillas y otros tipos de material genético para el futuro.

CLI. Del inglés, *Command Line Interface*, línea de comandos tradicional del sistema operativo que permite la ejecución de programas y funciones mediante la introducción de comandos de texto.

Cliente. Computadora u otro dispositivo informático que consume servicios, recursos o datos provenientes de un servidor.

Cloud(*Nube*). Es un entorno digital que virtualiza recursos de hardware (IaaS) y software, ofreciendo servicios para el desarrollo de aplicaciones (PaaS) y proporcionando, además, aplicaciones listas (SaaS) para uso de los usuarios. Este modelo permite abstraer la necesidad de los servidores físicos y otros elementos de infraestructura propios en una empresa, ofreciendo una solución más flexible económica, ya que se paga sólo por los recursos que se consumen.

DevOps. Cultura, metodología y conjunto de prácticas que combina el desarrollo de software (Dev) y las operaciones de IT (Ops) para mejorar la colaboración, la automatización y la eficiencia en la entrega de aplicaciones y servicios. Su objetivo principal es reducir el tiempo entre la creación de una idea y su implementación en producción, asegurando al mismo tiempo la calidad, estabilidad y seguridad del software.

Dominio. Dirección o nombre único que se utiliza para identificar sitios web o servicios en la red. Es una parte de lo que es la URL.

Firewall. Componente de seguridad informática que permite monitorear, filtrar y controlar el acceso y el tráfico de datos que entra o sale de una red, con el objetivo de proteger a los sistemas de accesos no autorizados, ataques y amenazas. Funciona como barrera entre redes confiables (Redes internas) y redes no confiables (Internet).

Frontend. Se refiere a la parte visual e interactiva de una aplicación o sitio web. Es la capa con la que los usuarios interactúan de manera directa con los datos y funcionalidades del sistema, a través de un navegador u otra interfaz gráfica.

GUI. Del inglés, Graphical User Interface, software que ofrece una solución visual para acceder a funcionalidades y con las que interactúan los usuarios de forma más amigable e intuitiva.

Hardware. Conjunto de componentes físicos que contiene una computadora u otros dispositivos informáticos. Es responsable de procesar el software para que el sistema realice tareas y funciones requeridas por el usuario.

Inventario. Registro ordenado y sistemático de los recursos que se encuentran dentro de una organización. Permite llevar la cantidad de bienes, materiales o productos que se encuentran en un sitio específico.

Kubernetes. Plataforma de código abierto diseñada para automatizar la implementación, gestión y escalado de aplicaciones basadas en contenedores.

Logs. Archivos generados automáticamente por sistemas, aplicaciones, servidores o dispositivos que documentan eventos, actividades y mensajes relacionados con su funcionamiento. Son esenciales para el monitoreo, diagnóstico y optimización de sistemas informáticos.

Plugin. Software adicional que se descarga y provee nuevas funcionalidades o personalización a un producto de software.

Proxy inverso. Servidor que se sitúa delante de los servidores web y reenvía las solicitudes del cliente (por ejemplo, el navegador web) a esos servidores web. Los proxies inversos suelen implementarse para ayudar a aumentar la seguridad, el rendimiento y la fiabilidad.

Servidor. Computadora u otro dispositivo que brinda servicios, recursos o datos a otras computadoras a través de la red.

Software. Conjunto de programas, aplicaciones y sistemas operativos que permiten a una computadora u otros dispositivos realizar tareas específicas. Se compone de instrucciones y datos que son procesados por el hardware para la ejecución de las funciones.

SSL. Del inglés, Secure Socket Layer, es un protocolo de seguridad utilizado para cifrar el tráfico entre un cliente y un servidor a través de internet.

SSH. Del inglés, Secure Shell, es un protocolo de red utilizado para acceder a un sistema remoto de forma segura a través de una red no segura, como internet.

Taxonomy(Taxonomía). Ciencia que se encarga de describir, clasificar y organizar a los seres vivos u otros elementos de acuerdo con sus características y relaciones.

Traefik. Implementación de proxy inverso moderno y de código abierto que está diseñado para facilitar la gestión del tráfico web en aplicaciones.

URL. Del inglés, Uniform Resource Locator, es la dirección específica que se utiliza para acceder a un recurso en particular en internet, como puede ser una página web, una imagen o cualquier otro archivo.

USDA. Del inglés, United States Department of Agriculture, Agencia gubernamental responsable de desarrollar y ejecutar políticas relacionadas con la agricultura, alimentación, los recursos naturales y el desarrollo rural en Estados Unidos.

VPN. Del inglés, Virtual Private Network (Red Virtual Privada), es una herramienta de ciberseguridad(Software) que permite establecer conexiones seguras y cifradas entre un cliente y un servidor remoto en una red privada. Su objetivo es garantizar la privacidad y la seguridad de los datos, evitando que sean interceptados mientras se navega por internet.

Bibliografía

Azure DevOps

<https://azure.microsoft.com/es-es/products/devops>

Documentación oficial Docker

<https://docs.docker.com/manuals/>

Documentación oficial JMeter

<https://jmeter.apache.org/>

¿En qué consisten los servidores proxy inversos?

<https://www.cloudflare.com/es-es/learning/cdn/glossary/reverse-proxy/>

¿Qué es SSL?

<https://www.cloudflare.com/es-es/learning/ssl/what-is-ssl/>

¿Qué es una VPN?

<https://latam.kaspersky.com/resource-center/definitions/what-is-a-vpn?srsIid=AfmBOorD6DuTBwPNchEpfw8gAVyGNrhzy9wq8Rf9OUTXfM9PU2Eovwtw>

GRIN Global Community Edition

<https://www.croptrust.org/work/information-systems/grin-global/>

Anexos

Bitácora de trabajo

Cronología

En esta sección se detalla lo ocurrido y que fue documentado en cada semana dentro de la bitácora de trabajo.

Semana 1

El proyecto dio inicio con una reunión en la que se presentó el equipo, el contexto actual y la iniciativa del convenio entre ambas instituciones para llevarlo a cabo. También se proporcionó el contacto con el representante del *software* y se solicitó una nueva cita con gente especializada del área de sistemas para poder comprender cuales son los recursos con los que cuentan, cuales son sus requerimientos a nivel seguridad y cómo continuar para llevar a cabo la instalación. Mientras tanto, se comenzó con la redacción del protocolo para entregar a la cátedra de Trabajo Final, además de definir un plan de ejecución con la estimación de las tareas a realizar, definir el problema a resolver y el impacto esperado de su implementación. También se plantearon posibles preguntas para la próxima reunión que tendría como foco principal determinar la factibilidad de la realización del proyecto.

Semanas 2-4

Durante las primeras semanas, el equipo se dedicó al estudio de la solución, revisando la documentación oficial y otras fuentes complementarias. Además, se

realizaron consultas al representante mediante correos electrónicos para resolver las dudas que surgieron. Se observó que existía una solución basada en contenedores, una tecnología innovadora y eficiente que se decidió implementar.

Semana 5

Posteriormente, se tuvo una reunión con el representante del *software* y el área de ciberseguridad. En esta sesión se abordaron temas claves como los protocolos de comunicación, los requerimientos recomendados para el sistema, los recursos con los que cuenta el INTA y las políticas de seguridad implementadas. Se determinó que la instalación se realizaría en el datacenter propio del INTA mediante el uso de un cliente *VPN*, accediendo al puerto 22 por *SSH* y que el INTA se encargaría de registrar los dominios necesarios. Aunque los puertos se encontraban cerrados hacia el exterior, se acordó abrir los dos puertos requeridos por la solución, según lo solicitado.

Semanas 6-9

Quedamos a la espera de la aprobación del INTA para que provean la infraestructura y sus credenciales para dar comienzo a la instalación. Mientras tanto, se continuó con la investigación y la realización de prácticas del uso del sistema. Esto último se logró gracias a que el representante brindó un servidor de prueba. También se llevó a cabo una reunión de capacitación en la que participaron clientes de toda latinoamérica, en la que se presentó la arquitectura del sistema, el modelo de datos y los flujos básicos de funcionamiento.

Semanas 10-13

Una vez provistos los accesos y la ip del servidor por parte del INTA, se descargó y configuró el cliente *VPN* solicitado, lo que permitió probar las credenciales para acceder al servidor mediante un cliente *SSH*. A partir de ahí, se comenzó con la instalación del sistema. La solución multicontenedor está compuesta por un contenedor que contiene el *frontend* y otro que contiene la *API*, que se comunica con la base de datos, cada uno con sus respectivas configuraciones para asegurar la comunicación entre sí. Además, la base de datos se configuró y se desplegó en versión contenedor para seguir el mismo enfoque. Se generaron los archivos de configuración con los datos correspondientes y se realizó el despliegue.

Los primeros intentos no fueron exitosos. El contenedor de la *API* no iniciaba debido a un error. Tras revisar los registros(*Logs*) y la configuración sin poder encontrar la causa, se intentó nuevamente sin modificar nada y logró desplegarse. El siguiente problema surgió al querer acceder a los dominios, ya que se obtenía un timeout en la respuesta, lo que resultó extraño. Para resolverlo, se decidió enviar un correo electrónico al representante del *software* para solicitar soporte.

Se solicitó enviar los *logs* de los contenedores y los archivos utilizados. En estos se alertó que se estaba implementando por defecto un proxy inverso llamado *Traefik*. Teniendo en cuenta que INTA ya implementa un proxy inverso, se determinó que esa sería la razón por la cual no se conseguía acceder de manera correcta. Para solucionar esto, se decidió quitar el *proxy inverso* de la configuración. Luego del redespliegue y verificar que los contenedores quedaran en estado listo, se pudo acceder exitosamente desde el navegador.

Con la solución GGCE implementada, se probó la conexión al servidor utilizando el cliente descargado previamente para acceder al servidor con datos de prueba provisto por el representante. Al configurar los datos del servidor en la herramienta, se pudo establecer conexión con el servidor, quedando así una parte considerable del proyecto ejecutada.

Semanas 14-15

Para finalizar las tareas relacionadas al servidor, se generó la documentación de instalación y administración del mismo, la cual se entregó junto con las credenciales a los responsables de su administración seleccionados según sus cargos y responsabilidades dentro de la institución.

Durante la reunión, se hicieron consultas sobre el crecimiento de la base de datos y se solicitaron los accesos para que el estudiante pueda acceder a los servicios en la nube donde se gestionan los proyectos internos del INTA.

Entre las tareas solicitadas se tienen:

- La suba de las imágenes de contenedores utilizadas por el sistema al registro de contenedores, con el objetivo de mantener un control centralizado de las imágenes empleadas.
- La generación de la documentación del proyecto dentro de la herramienta Azure DevOps para que el nuevo administrador comprenda el sistema y lo complemente con los documentos generados.
- La carga de los archivos utilizados en la solución para control y posibles implementaciones para mejorar la gestión del proyecto, por ejemplo,

automatización del proceso de despliegue mediante pipelines de integración continua.

Semanas 15-20

Debido a la decisión de postergar la etapa de instalación de los clientes en los nodos y las pruebas de rendimiento para poder estudiar con mayor profundidad el funcionamiento de la solución y el dominio del problema, se continuó con la documentación de instalación de los clientes.

Posteriormente, se hizo una extensa documentación del uso de la herramienta GGCE y también sobre el cliente. Ambas herramientas brindan similares funciones pero la diferencia radica en que el cliente permite la carga masiva de datos desde una hoja de cálculo, siendo una pieza clave para lograr una migración efectiva de los datos que actualmente se tienen.

Para continuar, se dio inicio a la etapa de capacitación en la que ya habiendo tenido el tiempo suficiente para realizar demostraciones, resolver consultas, entregar toda la documentación generada y brindar soporte para lograr utilizar la herramienta de manera eficaz.

Las instalaciones de los clientes para el grupo de la referente funcional presentaron algunos obstáculos, ya que requirió de los permisos de los administradores de sus estaciones de trabajo. Esto implicó realizar una solicitud mediante un sistema de tickets. Finalmente, los administradores completaron la instalación de los clientes, guiándose por la documentación de instalación entregada.

Por último, se llevaron a cabo las pruebas de rendimiento utilizando la herramienta JMeter, la cual permitió evaluar cómo responde el sistema ante escenarios atípicos. Los resultados obtenidos fueron positivos, demostrando la solidez y la eficiencia de la solución.