



REPUBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD NACIONAL EXPERIMENTAL PARA LAS
TELECOMUNICACION E INFORMATICA
VICERRECTORADO ACADEMICO
PROGRAMA NACIONAL DE FORMACION EN INGENIERIA EN
INFORMATICA

Modelo de Actores y Unidades

Autor: Br. Wrallean Brito - C.I: 24.774.283

Br. Orlando Lopez - C.I: 31.332.501

Br. Pedro Castro - C.I: 32.736.571

Br. Jean Aguiar - C.I: 30.831.319

Tutor: Prof.

Caracas, 16 agosto de 2025



Estrategias Para Validar Requisitos

1. Revisiones de Requisitos:

a) Revisiones por Pares:

- Involucrar a otros miembros del equipo para revisar los requisitos.
Esto ayuda a identificar inconsistencias y omisiones.
- Realizar reuniones periódicas donde se discutan los requisitos y se obtenga retroalimentación.

b) Revisiones con Partes Interesadas / Interesados Clave:

- Presentar los requisitos a los interesados para asegurarse de que reflejan sus necesidades.
- Utilizar un formato claro y conciso para facilitar la comprensión.

2. Entrevistas:

a) Entrevistas a Usuarios Finales:

- Realizar entrevistas con los usuarios finales para comprender sus necesidades y expectativas.
- Formular preguntas abiertas que permitan a los usuarios expresar sus opiniones y sugerencias.

b) Entrevistas con Expertos:

- Consultar a expertos en el área para obtener su perspectiva sobre la viabilidad y relevancia de los requisitos.



3. Prototipos:

a) Prototipos de Baja Fidelidad:

- Crear bocetos o wireframes que representen la interfaz y funcionalidad del sistema. Esto permite a los usuarios visualizar el producto y proporcionar retroalimentación temprana.

b) Prototipos de Alta Fidelidad:

- Desarrollar versiones más avanzadas del prototipo que incluyan interactividad. Esto ayuda a los usuarios a experimentar el sistema de manera más realista y a identificar problemas potenciales.

4. Pruebas de Usabilidad

a) Sesiones de Pruebas con Usuarios:

- Realizar pruebas de usabilidad donde los usuarios interactúan con el prototipo y se observa su comportamiento.
- Recoger datos sobre la facilidad de uso y la satisfacción del usuario.

5. Documentación y Seguimiento

a) Registro de Cambios:

- Mantener un registro de todos los cambios realizados en los requisitos a lo largo del proyecto. Esto ayuda a rastrear la evolución de los requisitos y a justificar decisiones.

b) Revisiones Continuas:

- Establecer un proceso de revisión continua a lo largo del ciclo de vida del proyecto para asegurar que los requisitos sigan siendo relevantes y se ajusten a las necesidades cambiantes.



Definición de Seguimientos y Control de Cambios

Git es una herramienta poderosa para el control de versiones que permite gestionar cambios en el código y la documentación de un proyecto de manera eficiente. A continuación, se describen las estrategias para implementar el seguimiento y control de cambios utilizando Git.

1. Establecimiento de un Repositorio Git

a) Creación del Repositorio:

- Iniciar un repositorio Git para el proyecto, ya sea en una plataforma como GitHub, GitLab o Bitbucket.
- Asegurarse de que todos los miembros del equipo tengan acceso al repositorio.

b) Estructura del Repositorio:

- Organizar el repositorio con una estructura de carpetas clara que separe el código, la documentación y otros recursos del proyecto.
- Incluir un archivo README que explique el propósito del proyecto y cómo contribuir.

2. Control de Cambios

a) Commits Frecuentes:

- Realizar commits frecuentes con mensajes claros y descriptivos que expliquen el propósito de cada cambio.
- Seguir una convención de mensajes de commit para facilitar la comprensión del historial de cambios.



b) Ramas (Branches):

- Utilizar ramas para desarrollar nuevas características o realizar cambios significativos sin afectar la rama principal.
- Nombrar las ramas de manera descriptiva.

3. Revisión de Cambios

a) Pull Requests:

- Implementar un proceso de Pull Requests para que los cambios en las ramas sean revisados por otros miembros del equipo antes de ser fusionados a la rama principal.
- Utilizar comentarios en los PR para discutir cambios y sugerir mejoras.

b) Revisiones de Código:

- Fomentar revisiones de código como parte del proceso de PR, donde los miembros del equipo pueden proporcionar retroalimentación sobre el código y los requisitos implementados.

4. Documentación de Cambios

a) Changelog:

- Mantener un archivo CHANGELOG.md en el repositorio que documente los cambios significativos en cada versión del proyecto.
- Incluir detalles sobre nuevas características, correcciones de errores y cambios en los requisitos.

b) Etiquetas (Tags):

- Utilizar etiquetas para marcar versiones específicas del proyecto, facilitando el seguimiento de cambios a lo largo del tiempo.



- Seguir un esquema de versionado semántico, por ejemplo v1.0.0, para las etiquetas.

5. Seguimiento del Progreso

a) Historial de Commits:

- Utilizar el historial de commits de Git para rastrear cambios y entender la evolución del proyecto.
- Ejecutar comandos como git log para ver el historial de cambios y git diff para comparar diferencias entre versiones.



Identificación de Riesgos y Estrategias de Mitigación

1. Riesgos Técnicos

a) Riesgo: Fallos en la Tecnología

- **Descripción:** Dependencia de tecnologías que pueden fallar o no cumplir con las expectativas.
- **Mitigación:**
 - Realizar pruebas exhaustivas de las tecnologías seleccionadas antes de su implementación.
 - Mantener una lista de tecnologías alternativas que puedan ser utilizadas en caso de fallos.

b) Riesgo: Integración de Sistemas

- **Descripción:** Dificultades al integrar diferentes sistemas o componentes del proyecto.
- **Mitigación:**
 - Planificar y realizar pruebas de integración desde las primeras etapas del desarrollo.
 - Documentar claramente las interfaces y protocolos de comunicación entre sistemas.

2. Riesgos de Requisitos

a) Riesgo: Cambios en los Requisitos:

- **Descripción:** Cambios inesperados en los requisitos que pueden afectar el alcance y el cronograma del proyecto.



- **Mitigación:**

- Establecer un proceso formal de control de cambios para gestionar las solicitudes de modificación de requisitos.
- Involucrar a los interesados en revisiones periódicas para asegurar que los requisitos se mantengan alineados con sus necesidades.

b) Riesgo: Requisitos Incompletos o Mal Entendidos:

- **Descripción:** Requisitos que no están bien definidos o que son malinterpretados por el equipo.
- **Mitigación:**
 - Realizar sesiones de validación de requisitos con los interesados para aclarar y confirmar los requisitos.
 - Utilizar prototipos y revisiones para asegurar que los requisitos sean comprendidos correctamente.

3. Riesgos de Gestión del Proyecto

a) Riesgo: Retrasos en el Cronograma:

- **Descripción:** El proyecto puede no cumplir con los plazos establecidos debido a diversos factores.
- **Mitigación:**
 - Establecer un cronograma realista con márgenes de tiempo para imprevistos.
 - Realizar revisiones periódicas del progreso y ajustar el cronograma según sea necesario.



b) Riesgo: Falta de Recursos:

- **Descripción:** Insuficiencia de recursos humanos, técnicos o financieros para completar el proyecto.
- **Mitigación:**
 - Planificar y asignar recursos de manera efectiva desde el inicio del proyecto.
 - Identificar y asegurar recursos adicionales que puedan ser necesarios a lo largo del proyecto.

4. Riesgos de Comunicación

a) Riesgo: Falta de Comunicación entre el Equipo:

- **Descripción:** La falta de comunicación puede llevar a malentendidos y errores en el desarrollo.
- **Mitigación:**
 - Establecer canales de comunicación claros y regulares, como reuniones diarias o semanales.
 - Utilizar herramientas de gestión de proyectos para mantener a todos los miembros del equipo informados sobre el progreso y los cambios.

b) Riesgo: Desalineación con los Interesados:

- **Descripción:** Los interesados pueden no estar alineados con el progreso o los resultados del proyecto.
- **Mitigación:**
 - Realizar reuniones periódicas de seguimiento con los interesados para mantenerlos informados y obtener su retroalimentación.



- Documentar y comunicar claramente los objetivos y el progreso del proyecto.

5. Riesgos de Calidad

a) Riesgo: Baja Calidad del Producto Final:

- **Descripción:** El producto final puede no cumplir con los estándares de calidad esperados.
- **Mitigación:**
 - Implementar un proceso de control de calidad que incluya pruebas regulares y revisiones del producto.
 - Fomentar una cultura de calidad dentro del equipo, donde todos sean responsables de la calidad del trabajo.



Propuesta de Índice de Documento Final

1. Introducción

1.1. Propósito del Proyecto

1.2. Justificación

1.3. Alcance General

1.4. Usuarios Directos

1.5. Servicios Clave

1.6. Ámbito de Operación

2. Modelado de Dominio y Objetivos del Negocio

2.1. Introducción

2.2. Contexto Universitario

2.3. Objetivos del Negocio

2.3.1. Automatización de Asignación de Tareas con IA

2.3.2. Facilitar la Colaboración en Tiempo Real

2.3.3. Centralización de la Gestión de Proyectos Académicos

2.3.4. Mejora de la Trazabilidad de Carga de Trabajo

2.3.5. Optimización de la Distribución de Carga de Trabajo

2.3.6. Experiencia Fluida y Segura

2.3.7. Reducción de la Dependencia de Planificación Manual

2.4. Diagrama de Objetivos del Sistema

3. Actores y Organización

3.1. Identificación de los Actores

3.1.1. Administrador del Sistema



3.1.2. Líder del Proyecto

3.1.3. Usuario Regular

3.1.4. Colaborador

3.1.5. Visitante

3.2. Organigrama de Actores

4. Modelo de Actores y Unidades

4.1. Objetos del Sistema y sus Atributos Relevantes

4.1.1. Objetos de Gestión de Usuarios

4.1.1.1. Usuario

4.1.1.2. Rol

4.1.1.3. Sesión

4.1.2. Objetos de Gestión de Proyectos

4.1.2.1. Proyecto

4.1.2.2. Tablero

4.1.2.3. Plantilla

4.1.3. Objetos de Gestión de Tareas

4.1.3.1. Tarea

4.1.3.2. Comentario

4.1.4. Objetos de Inteligencia Artificial

4.1.4.1. Perfil de Habilidades

4.1.4.2. Chatbot de IA

4.1.5. Objetos de Colaboración

4.1.5.1. Notificación

4.1.5.2. Archivo Adjunto



4.1.6. Objetos de Auditoría

4.1.6.1. Historial de Cambios

4.2. Diagrama de Clases

4.3. Matriz Procesos vs. Objetos/Clases

5. Procesos del Negocio y Apoyo de la Aplicación

5.1. Identificación y Descripción de Procesos de Negocio

5.1.1. Gestión de Usuarios

5.1.2. Gestión de Proyectos

5.1.3. Gestión de Tareas

5.1.4. Procesamiento de Inteligencia Artificial (IA)

5.1.5. Gestión de Colaboración

5.2. Determinación de Procesos Apoyados por la Aplicación

5.3. Diagrama de Procesos UML (Actividad o Flujo)

5.4. Matriz de Actores vs. Procesos

6. Requisitos del Sistema

6.1. Requisitos Funcionales (RF)

6.1.1. Prioridad Alta

6.1.2. Prioridad Media

6.1.3. Prioridad Baja

6.2. Requisitos No Funcionales (RNF)

6.2.1. Prioridad Alta

6.2.2. Prioridad Media

6.2.3. Prioridad Baja

6.3. Requisitos Funcionales Clave (Plantilla Volere)



6.4. Requisitos No Funcionales Clave (Plantilla Volere)

7. Escenario y Diagramas

7.1. Contexto del Escenario

7.2. Diagrama de Casos de Uso

7.3. Diagrama de Clases (del Escenario)

8. Validación, Seguimiento y Estructura Final

8.1. Estrategias para Validar Requisitos

8.1.1. Revisiones de Requisitos

8.1.2. Entrevistas

8.1.3. Prototipos

8.1.4. Pruebas de Usabilidad

8.1.5. Documentación y Seguimiento

8.2. Definición de Seguimiento y Control de Cambios

8.2.1. Establecimiento de un Repositorio Git

8.2.2. Control de Cambios

8.2.3. Revisión de Cambios

8.2.4. Documentación de Cambios

8.2.5. Seguimiento del Progreso

8.3. Identificación de Riesgos y Estrategias de Mitigación

8.3.1. Riesgos Técnicos

8.3.2. Riesgos de Requisitos

8.3.3. Riesgos de Gestión del Proyecto

8.3.4. Riesgos de Comunicación

8.3.5. Riesgos de Calidad