



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

**Propuesta del Proyecto Análisis del Uso de
Herramientas Tecnológicas**

Curso: Inteligencia de Negocios

Docente: Mag. Patrick Cuadros Quiroga

Integrantes:

Hernández Cruz, Angel Gadiel (2021070017)

Paja de la Cruz, Piero Alexander (2020067576)

**Tacna – Perú
2025**

Proyecto

Análisis del Uso de Herramientas Tecnológicas en los proyectos de Estudiantes Sistemas UPT, Tacna, 2025

Presentado por:

Hernández Cruz, Angel Gadiel

Paja de la Cruz, Piero Alexander

Estudiantes de Ingeniería de Sistemas

Junio 2025

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	AHC, PPC	PCQ	ARV	18/03/2025	Versión Original

Tabla de contenido

Resumen Ejecutivo	3
I Propuesta narrativa	4
1. Planteamiento del Problema.....	4
2. Justificación del proyecto	4
3. Objetivo general	5
4. Beneficios	6
5. Alcance	7
6. Requerimientos del sistema	7
7. Restricciones	7
8. Supuestos.....	7
9. Resultados esperados	8
10. Metodología de implementación	8
11. Actores claves	9
12. Papel y responsabilidades del personal	10
13. Plan de monitoreo y evaluación	12
14. Cronograma del proyecto	15
15. Hitos de entregables	17
II Presupuesto	18
1. Planteamiento de aplicación del presupuesto	18
2. Presupuesto	18
3. Análisis de Factibilidad	18

4. Evaluación Financiera.....	18
Anexo 01 – Requerimientos del Sistema <i>{nombre del sistema}</i>	19

Resumen Ejecutivo

El proyecto "Análisis del Uso de Herramientas Tecnológicas en los proyectos de Estudiantes Sistemas UPT" surge de la necesidad de comprender y optimizar el uso de tecnologías en el entorno académico de la Universidad Privada de Tacna.

Los estudiantes de Ingeniería de Sistemas enfrentan dificultades para utilizar eficientemente las herramientas tecnológicas disponibles, lo que impacta negativamente en la calidad de sus proyectos y su preparación profesional. Este análisis permitirá identificar patrones de uso, herramientas más populares y áreas de mejora en la formación tecnológica.

El proyecto tiene una duración de 5 meses con un presupuesto total de S/5,250 y beneficiará directamente a estudiantes, docentes y administradores de la institución. Se implementará mediante recolección automatizada de datos desde repositorios Git, análisis con herramientas de código abierto y generación de dashboards interactivos para la toma de decisiones.

Los resultados esperados incluyen mejoras en la formación tecnológica, optimización del uso de recursos y alineación de competencias estudiantiles con las demandas del mercado laboral actual.

I. Propuesta narrativa

1. Planteamiento del Problema

Los estudiantes de la Universidad Privada de Tacna, específicamente de la Escuela de Ingeniería de Sistemas, no aprovechan al máximo las herramientas tecnológicas disponibles para sus proyectos académicos. Esta problemática se manifiesta en varios aspectos críticos:

- **Falta de estandarización:** No existe un criterio uniforme para la selección y uso de herramientas tecnológicas en los proyectos académicos.
- **Brechas en capacitación:** Los estudiantes carecen de formación adecuada para utilizar eficientemente las tecnologías disponibles.
- **Acceso desigual a recursos:** Algunos estudiantes enfrentan limitaciones para acceder a herramientas tecnológicas fuera del entorno universitario.
- **Desconocimiento de tendencias:** Falta de información sobre cuáles son las herramientas más demandadas en el mercado laboral actual.

Esta situación genera un impacto negativo en la calidad de los proyectos, el aprendizaje de los estudiantes y su preparación para el mercado laboral, creando una brecha entre la formación académica y las necesidades profesionales reales.

2. Justificación del proyecto

El proyecto se justifica por las siguientes razones fundamentales:

Académica: Es necesario comprender cómo los estudiantes utilizan las herramientas tecnológicas para identificar fortalezas y debilidades en su formación, permitiendo mejoras curriculares basadas en evidencia.

Tecnológica: El análisis permitirá identificar las tecnologías más relevantes y efectivas, optimizando la inversión en recursos tecnológicos de la universidad.

Profesional: Los resultados ayudarán a alinear la formación académica con las demandas del mercado laboral, mejorando la empleabilidad de los graduados.

Institucional: La universidad podrá tomar decisiones informadas sobre políticas tecnológicas, capacitación docente y modernización de laboratorios.

Social: Contribuirá a reducir la brecha digital y tecnológica, promoviendo un uso más equitativo y eficiente de las herramientas disponibles.

3. Objetivo general

Analizar el uso de herramientas tecnológicas en los proyectos académicos de los estudiantes de Ingeniería de Sistemas de la Universidad Privada de Tacna para identificar patrones de uso, evaluar su impacto en el rendimiento académico y proponer mejoras en la formación tecnológica.

Objetivos Específicos:

1. **Evaluar la distribución de lenguajes de programación** utilizados en los proyectos estudiantiles para identificar tendencias y preferencias.
2. **Identificar frameworks y librerías más utilizados** por los estudiantes en sus desarrollos académicos.
3. **Analizar el uso de bases de datos** (relacionales y NoSQL) en los proyectos para determinar competencias en gestión de datos.
4. **Investigar la implementación de herramientas CI/CD** en el entorno académico y su impacto en la calidad de los proyectos.
5. **Generar recomendaciones** para mejorar la formación tecnológica basadas en los hallazgos del análisis.

4. Beneficios

Para los Estudiantes:

- Mejor comprensión de las herramientas tecnológicas más demandadas
- Orientación para elegir tecnologías relevantes en sus proyectos
- Mejora en la calidad de sus desarrollos académicos
- Mayor preparación para el mercado laboral

Para los Docentes:

- Información actualizada sobre tendencias tecnológicas estudiantiles
- Herramientas para mejorar la planificación curricular
- Identificación de áreas donde se requiere capacitación adicional
- Recursos para orientar mejor a los estudiantes

Para la Universidad:

- Optimización de la inversión en recursos tecnológicos
- Mejora en el posicionamiento académico
- Alineación de la oferta educativa con las demandas del mercado
- Datos para la toma de decisiones estratégicas

Para la Industria:

- Graduados mejor preparados tecnológicamente
- Reducción de tiempo de adaptación en nuevos empleos
- Alineación entre competencias académicas y necesidades profesionales

5. Alcance

Incluye:

- Análisis de repositorios Git de proyectos estudiantiles
- Evaluación de lenguajes de programación, frameworks, librerías, bases de datos y herramientas CI/CD
- Estudiantes de todos los niveles de la Escuela de Ingeniería de Sistemas UPT
- Proyectos académicos desarrollados entre 2023-2025
- Generación de dashboards interactivos con Power BI
- Recomendaciones para mejora curricular

No Incluye:

- Proyectos de otras escuelas profesionales
- Análisis de proyectos privados sin autorización
- Datos personales sensibles de estudiantes
- Evaluación de rendimiento individual de estudiantes
- Implementación de cambios curriculares (solo recomendaciones)

Limitaciones:

- Análisis limitado a repositorios públicos o con autorización
- Dependencia de la calidad de documentación en los proyectos
- Tiempo limitado a 5 meses de desarrollo

6. Requerimientos del sistema

Requerimientos Funcionales:

- **RF01:** El sistema debe extraer metadatos de repositorios Git automáticamente
- **RF02:** El sistema debe clasificar lenguajes de programación por proyecto
- **RF03:** El sistema debe identificar frameworks utilizados en cada proyecto
- **RF04:** El sistema debe detectar librerías empleadas por contexto de lenguaje
- **RF05:** El sistema debe identificar una base de datos principal por proyecto
- **RF06:** El sistema debe reconocer herramientas CI/CD implementadas
- **RF07:** El sistema debe generar reportes estadísticos visuales
- **RF08:** El sistema debe permitir filtros por periodo académico y curso

Requerimientos No Funcionales:

- **RNF01:** El sistema debe procesar más de 500 repositorios sin fallar
- **RNF02:** El tiempo de procesamiento no debe exceder 30 segundos por repositorio
- **RNF03:** Los reportes deben generarse en menos de 2 minutos
- **RNF04:** La disponibilidad del sistema debe ser del 99.5%
- **RNF05:** El sistema debe ser compatible con navegadores modernos
- **RNF06:** Los datos deben estar encriptados con AES-256

7. Restricciones

Temporales:

- El proyecto debe completarse en máximo 5 meses
- Entregables mensuales obligatorios
- Presentación final en julio 2025

Tecnológicas:

- Uso prioritario de herramientas de código abierto
- Compatibilidad solo con repositorios Git públicos
- Limitaciones de la API de GitHub (5000 requests/hora)

Presupuestarias:

- Presupuesto máximo de S/5,250
- No se pueden contratar servicios externos de desarrollo
- Limitación en licencias de software comercial

Legales:

- Cumplimiento estricto de la Ley N° 29733 (Protección de Datos Personales)
- Autorización requerida para análisis de proyectos no públicos
- Respeto a los términos de uso de GitHub API

Operativas:

- Dependencia de la infraestructura tecnológica de la UPT
- Disponibilidad limitada de laboratorios especializados
- Acceso a internet estable requerido

8. Supuestos

Técnicos:

- Los estudiantes utilizan GitHub como plataforma principal para sus proyectos
- Los repositorios contienen documentación mínima (README.md)
- La infraestructura de la UPT soportará las cargas de procesamiento
- Las herramientas de código abierto seleccionadas mantendrán su disponibilidad

Organizacionales:

- La universidad brindará el apoyo institucional necesario
- Los docentes colaborarán proporcionando información sobre proyectos
- Los estudiantes darán consentimiento para el análisis de sus proyectos públicos
- El departamento de TI proporcionará el soporte técnico requerido

Metodológicos:

- Los metadatos de los repositorios reflejan accuradamente las tecnologías utilizadas
- Los patrones identificados serán representativos del comportamiento general
- La muestra de proyectos será suficiente para obtener conclusiones válidas

Externos:

- GitHub mantendrá su API disponible durante el proyecto
- No habrá cambios significativos en las políticas de acceso a datos
- Las herramientas tecnológicas analizadas mantendrán su relevancia

9. Resultados esperados

Productos Entregables:

- Dashboard interactivo con visualizaciones de uso de tecnologías
- Informe técnico completo con análisis estadístico
- Base de datos normalizada con información de proyectos
- Documento de recomendaciones para mejora curricular
- Presentación ejecutiva para autoridades universitarias

Indicadores de Éxito:

- Análisis de al menos 400 repositorios de proyectos estudiantiles
- Identificación de las 10 tecnologías más utilizadas por categoría
- Generación de métricas de adopción tecnológica por año académico
- Propuesta de 5 recomendaciones concretas para mejora curricular
- Satisfacción del 90% de usuarios finales con los dashboards

Impacto Esperado:

- Mejora del 20% en la alineación tecnológica curricular

- Reducción del 30% en tiempo de análisis manual de tendencias
- Incremento del 15% en la satisfacción estudiantil con recursos tecnológicos
- Optimización del 25% en la inversión en nuevas tecnologías

Transferencia de Conocimiento:

- Capacitación a 5 docentes en el uso del sistema
- Documentación técnica completa para mantenimiento
- Manual de usuario para administradores académicos
- Procedimientos para actualización periódica de datos

10. Metodología de implementación

Fase 1: Análisis y Diseño (Mes 1)

- Levantamiento detallado de requerimientos
- Análisis de factibilidad técnica y operativa
- Diseño de arquitectura del sistema
- Definición de casos de uso principales
- Elaboración del plan de proyecto detallado

Fase 2: Desarrollo del ETL (Mes 2)

- Configuración del entorno de desarrollo
- Implementación de extracción de datos desde GitHub API
- Desarrollo de algoritmos de clasificación tecnológica
- Creación de base de datos analítica
- Pruebas unitarias de componentes

Fase 3: Análisis y Procesamiento (Mes 3)

- Recolección masiva de datos de repositorios
- Procesamiento y limpieza de información
- Aplicación de algoritmos de análisis estadístico
- Validación de datos y corrección de inconsistencias
- Generación de métricas preliminares

Fase 4: Visualización y Dashboards (Mes 4)

- Desarrollo de dashboards en Power BI
- Implementación de filtros interactivos
- Diseño de visualizaciones optimizadas
- Integración con fuentes de datos actualizadas
- Pruebas de usabilidad con usuarios finales

Fase 5: Validación y Despliegue (Mes 5)

- Pruebas integrales del sistema completo
- Validación con stakeholders académicos

- Refinamiento basado en retroalimentación
- Documentación final y capacitación
- Presentación de resultados y entrega formal

Metodología de Desarrollo:

- Metodología ágil con entregas incrementales
- Reuniones semanales de seguimiento
- Revisiones quinquenales con stakeholders
- Documentación continua del proceso
- Control de versiones con Git

11. Actores claves

Patrocinador del Proyecto:

- **Universidad Privada de Tacna** - Escuela de Ingeniería de Sistemas
- Proporciona recursos, infraestructura y apoyo institucional

Stakeholders Académicos:

- **Mag. Patrick Cuadros Quiroga** - Docente supervisor y asesor técnico
- **Director de Escuela de Sistemas** - Validador de requerimientos académicos
- **Coordinadores de curso** - Proveedores de contexto curricular
- **Docentes de programación** - Usuarios finales del sistema

Equipo de Desarrollo:

- **Hernández Cruz, Angel Gadiel** - Desarrollador principal y analista de datos
- **Paja de la Cruz, Piero Alexander** - Desarrollador de backend y especialista en ETL

Usuarios Finales:

- **Estudiantes de Ingeniería de Sistemas** - Beneficiarios directos
- **Administradores académicos** - Usuarios de dashboards y reportes
- **Personal de TI** - Responsables de mantenimiento técnico

Stakeholders Externos:

- **Departamento de TI UPT** - Soporte de infraestructura
- **Biblioteca UPT** - Apoyo en investigación y documentación
- **Industria local** - Beneficiarios indirectos con mejor preparación de graduados

Reguladores:

- **Autoridad Nacional de Protección de Datos Personales** - Cumplimiento legal
- **Comité de Ética UPT** - Validación ética del proyecto

12. Papel y responsabilidades del personal

Hernández Cruz, Angel Gadiel - Líder de Proyecto y Desarrollador Principal

Responsabilidades:

- Liderar la planificación y ejecución general del proyecto
- Desarrollar los componentes de extracción de datos (GitHub API)
- Implementar algoritmos de análisis y clasificación tecnológica
- Coordinar con stakeholders académicos y técnicos
- Supervisar la calidad y consistencia de los entregables

Funciones específicas:

- Configuración y mantenimiento del entorno de desarrollo
- Desarrollo del script de scraping principal (scrap_pequeno.py)
- Implementación de la lógica de clasificación de tecnologías
- Diseño y creación de la base de datos analítica
- Gestión de riesgos y resolución de problemas técnicos

Dedicación: 25 horas/semana durante 5 meses

Paja de la Cruz, Piero Alexander - Desarrollador de Backend y Especialista en ETL

Responsabilidades:

- Desarrollar los procesos de transformación y carga de datos (ETL)
- Implementar la arquitectura de base de datos en Azure SQL
- Crear y mantener los dashboards en Power BI
- Realizar pruebas de calidad y validación de datos
- Documentar procesos técnicos y procedimientos

Funciones específicas:

- Diseño del esquema de base de datos (schema.sql)
- Implementación de procedimientos de carga incremental
- Desarrollo de visualizaciones interactivas en Power BI
- Configuración de infraestructura en Azure (Terraform)
- Elaboración de manuales técnicos y de usuario

Dedicación: 25 horas/semana durante 5 meses

Mag. Patrick Cuadros Quiroga - Supervisor Académico y Asesor Técnico

Responsabilidades:

- Supervisar el progreso académico y técnico del proyecto
- Validar requerimientos y especificaciones del sistema
- Revisar y aprobar entregables principales
- Proporcionar orientación metodológica y académica

- Facilitar comunicación con autoridades universitarias

Funciones específicas:

- Reuniones semanales de seguimiento y orientación
- Revisión de documentos técnicos y académicos
- Validación de metodologías de análisis empleadas
- Apoyo en la interpretación de resultados estadísticos
- Presentación de avances a autoridades académicas

Dedicación: 4 horas/semana durante 5 meses

Director de Escuela de Ingeniería de Sistemas - Stakeholder Académico Principal

Responsabilidades:

- Definir objetivos académicos y expectativas del proyecto
- Validar relevancia y aplicabilidad de resultados
- Autorizar acceso a información institucional necesaria
- Evaluar impacto y beneficios para la escuela
- Facilitar implementación de recomendaciones

Funciones específicas:

- Definición de criterios de éxito académico
- Validación de requerimientos curriculares
- Autorización para análisis de proyectos estudiantiles
- Evaluación de propuestas de mejora curricular
- Comunicación con docentes y estudiantes

Dedicación: 2 horas/semana durante 5 meses

Departamento de TI UPT - Soporte Técnico e Infraestructura

Responsabilidades:

- Proporcionar acceso a infraestructura tecnológica necesaria
- Brindar soporte técnico para resolución de problemas
- Facilitar acceso a redes y servicios universitarios
- Validar aspectos de seguridad y cumplimiento
- Apoyar en el despliegue final del sistema

Funciones específicas:

- Configuración de accesos a redes universitarias
- Soporte en configuración de servidores y bases de datos
- Validación de políticas de seguridad informática
- Asistencia en resolución de problemas técnicos
- Capacitación en herramientas institucionales

Dedicación: 3 horas/semana durante 5 meses

Matriz de Responsabilidades (RACI):

Actividad	Angel H.	Piero P.	Prof. Cuadros	Director	TI UPT
Planificación del proyecto	R	A	C	I	I
Desarrollo de ETL	A	R	C	I	S
Creación de dashboards	C	R	C	I	S
Validación académica	I	I	R	A	I
Documentación técnica	R	A	C	I	I
Presentación final	R	A	R	A	I

Legenda: R=Responsable, A=Aprobador, C=Consultado, I=Informado, S=Soporte

13. Plan de monitoreo y evaluación

Framework de Monitoreo:

El proyecto implementará un sistema de monitoreo continuo basado en indicadores cuantitativos y cualitativos que permitan evaluar el progreso, identificar desviaciones y tomar acciones correctivas oportunas.

Indicadores de Progreso Técnico:

Indicador	Meta	Frecuencia	Responsable
Repositorios analizados	400+ proyectos	Semanal	Angel H.
Precisión en clasificación	≥95% exactitud	Quincenal	Piero P.
Disponibilidad del sistema	≥99.5% uptime	Diaria	Ambos
Cobertura de tecnologías	50+ herramientas	Mensual	Angel H.
Tiempo de procesamiento	<30 seg/repo	Semanal	Piero P.

Indicadores de Calidad Académica:

Indicador	Meta	Frecuencia	Responsable
Satisfacción de stakeholders	≥90% satisfacción	Mensual	Prof. Cuadros
Relevancia de análisis	100% requerimientos	Quincenal	Director
Calidad de visualizaciones	≥8/10 puntuación	Mensual	Piero P.

Utilidad de recomendaciones	5+ propuestas válidas	Al final	Angel H.
-----------------------------	-----------------------	----------	----------

Métodos de Evaluación:

Evaluación Técnica:

- Pruebas automatizadas de componentes del sistema
- Validación cruzada de datos con fuentes externas
- Métricas de rendimiento y optimización
- Auditorías de código y documentación técnica

Evaluación Académica:

- Reuniones de revisión con stakeholders
- Encuestas de satisfacción a usuarios finales
- Validación de resultados con expertos académicos
- Comparación con estudios similares en la literatura

Evaluación de Gestión:

- Seguimiento de cronograma y hitos
- Control de presupuesto y recursos
- Evaluación de riesgos identificados
- Medición de cumplimiento de entregables

Cronograma de Evaluaciones:

Evaluaciones Semanales:

- Reunión de equipo de desarrollo (lunes)
- Revisión de métricas técnicas (miércoles)
- Comunicación de avances a supervisor (viernes)

Evaluaciones Mensuales:

- Presentación formal a stakeholders
- Evaluación de satisfacción de usuarios
- Revisión y actualización de riesgos
- Ajuste de planificación si es necesario

Evaluaciones de Hitos:

- Evaluación integral al completar cada fase
- Validación de entregables principales
- Autorización para continuar a siguiente fase
- Documentación de lecciones aprendidas

Herramientas de Monitoreo:

Dashboard de Proyecto:

- Panel en tiempo real con métricas clave
- Alertas automáticas por desviaciones
- Visualización de progreso por componente
- Histórico de tendencias y comparaciones

Reportes Automatizados:

- Reporte semanal de avance técnico
- Informe mensual ejecutivo para dirección
- Alertas de riesgo por email
- Logs detallados de actividades del sistema

Gestión de Riesgos:

Identificación Continua:

- Reuniones semanales de identificación de riesgos
- Matriz de riesgos actualizada mensualmente
- Escalación automática de riesgos críticos
- Evaluación de impacto y probabilidad

Mitigación Proactiva:

- Planes de contingencia predefinidos
- Recursos de respaldo identificados
- Procedimientos de escalación claros
- Comunicación inmediata a stakeholders

Plan de Acciones Correctivas:

Desviaciones Menores (Impacto < 15%):

- Ajustes por parte del equipo de desarrollo
- Notificación a supervisor académico
- Documentación en bitácora del proyecto
- Seguimiento en próxima reunión semanal

Desviaciones Mayores (Impacto 15-30%):

- Reunión de emergencia con stakeholders
- Análisis de causas raíz
- Redefinición de plan de trabajo
- Autorización formal para continuar

Desviaciones Críticas (Impacto > 30%):

- Escalación a autoridades universitarias
- Evaluación de viabilidad del proyecto

- Posible redefinición de alcance
- Comunicación formal a todos los stakeholders

14. Cronograma del proyecto

Planificación General (5 meses - Marzo a Julio 2025):

Fase	Duración	Inicio	Fin	Hitos Principales
Fase 1: Análisis y Diseño	4 semanas	03/03/2025	28/03/2025	Documento de diseño aprobado
Fase 2: Desarrollo ETL	4 semanas	31/03/2025	25/04/2025	Sistema de extracción funcional
Fase 3: Análisis y Procesamiento	4 semanas	28/04/2025	23/05/2025	Base de datos poblada
Fase 4: Visualización	4 semanas	26/05/2025	20/06/2025	Dashboards implementados
Fase 5: Validación y Entrega	3 semanas	23/06/2025	11/07/2025	Proyecto finalizado

Cronograma Detallado por Fase:

FASE 1: ANÁLISIS Y DISEÑO (03/03 - 28/03/2025)

Semana	Actividades	Responsable	Entregables
1	Levantamiento de requerimientos detallados	Angel, Piero	Documento SRS
1	Análisis de factibilidad técnica y económica	Angel, Piero	Informe de factibilidad
2	Diseño de arquitectura del sistema	Angel	Documento de arquitectura
2	Definición de casos de uso principales	Piero	Diagramas UML
3	Diseño de base de datos analítica	Piero	Esquema de BD
3	Planificación detallada del proyecto	Angel	Plan de proyecto
4	Revisión y aprobación de documentos	Prof. Cuadros	Documentos aprobados

FASE 2: DESARROLLO ETL (31/03 - 25/04/2025)

Semana	Actividades	Responsable	Entregables
5	Configuración del entorno de desarrollo	Ambos	Ambiente configurado
5	Implementación de conexión a GitHub API	Angel	Módulo de conexión
6	Desarrollo de extractor de metadatos	Angel	Script de extracción
6	Implementación de clasificador de tecnologías	Angel	Algoritmos de clasificación
7	Creación de base de datos en Azure	Piero	BD desplegada
7	Desarrollo de procesos ETL	Piero	Pipeline ETL
8	Pruebas unitarias y validación	Ambos	Tests pasando

FASE 3: ANÁLISIS Y PROCESAMIENTO (28/04 - 23/05/2025)

Semana	Actividades	Responsable	Entregables
9	Recolección masiva de datos de repositorios	Angel	Dataset principal
9	Procesamiento y limpieza de información	Angel	Datos limpios
10	Aplicación de algoritmos de análisis	Angel	Métricas calculadas
10	Validación de datos y corrección	Piero	Datos validados
11	Generación de estadísticas preliminares	Piero	Reportes estadísticos
11	Análisis de patrones y tendencias	Angel	Análisis de tendencias
12	Documentación de hallazgos	Ambos	Informe de análisis

FASE 4: VISUALIZACIÓN Y DASHBOARDS (26/05 - 20/06/2025)

Semana	Actividades	Responsable	Entregables
13	Desarrollo de dashboards en Power BI	Piero	Dashboards básicos
13	Implementación de filtros interactivos	Piero	Filtros funcionales
14	Diseño de visualizaciones optimizadas	Piero	Gráficos avanzados
14	Integración con fuentes de datos	Angel	Conexiones activas
15	Pruebas de usabilidad con usuarios	Ambos	Feedback recogido

15	Refinamiento basado en retroalimentación	Piero	Dashboards refinados
16	Documentación de dashboards	Piero	Manual de usuario

FASE 5: VALIDACIÓN Y ENTREGA (23/06 - 11/07/2025)

Semana	Actividades	Responsable	Entregables
17	Pruebas integrales del sistema	Ambos	Sistema validado
17	Validación con stakeholders académicos	Prof. Cuadros	Validación formal
18	Elaboración de documentación final	Ambos	Documentación completa
18	Capacitación a usuarios finales	Piero	Usuarios capacitados
19	Presentación final y entrega	Ambos	Proyecto entregado

Dependencias Críticas:

- Acceso a GitHub API (requerido desde semana 5)
- Infraestructura Azure disponible (requerida desde semana 7)
- Aprobación de stakeholders (requerida en semanas 4, 8, 12, 16)
- Disponibilidad de datos de prueba (requerida desde semana 9)

Recursos por Fase:

- Fase 1: 2 desarrolladores + 1 supervisor (80 horas/semana)
- Fase 2: 2 desarrolladores + soporte TI (60 horas/semana)
- Fase 3: 2 desarrolladores (50 horas/semana)
- Fase 4: 1 desarrollador principal + diseñador (40 horas/semana)
- Fase 5: 2 desarrolladores + stakeholders (50 horas/semana)

15. Hitos de entregables

Hito 1: Documentación de Proyecto Aprobada (28/03/2025)

Entregables:

- Documento de Especificación de Requerimientos de Software (SRS)
- Informe de Factibilidad Técnica y Económica
- Documento de Arquitectura de Software (SAD)
- Plan de Proyecto Detallado con cronograma
- Documento de Visión del Proyecto

Criterios de Aceptación:

- Todos los documentos revisados y aprobados por el supervisor académico
- Requerimientos funcionales y no funcionales claramente definidos
- Arquitectura técnica validada por el departamento de TI
- Plan de proyecto con cronograma realista y recursos asignados
- Presupuesto aprobado por autoridades universitarias

Riesgos Asociados:

- Retrasos en aprobaciones institucionales
- Cambios en requerimientos por parte de stakeholders
- Subestimación de complejidad técnica

Hito 2: Sistema ETL Funcional (25/04/2025)

Entregables:

- Módulo de conexión a GitHub API implementado
- Script de extracción de metadatos funcional (scrap_pequeno.py)
- Algoritmos de clasificación de tecnologías validados
- Base de datos Azure SQL desplegada y configurada
- Pipeline ETL completo y probado
- Documentación técnica del sistema

Criterios de Aceptación:

- Capacidad de procesar al menos 100 repositorios sin errores
- Precisión de clasificación tecnológica $\geq 90\%$
- Tiempo de procesamiento < 45 segundos por repositorio
- Base de datos poblada con datos de prueba
- Todos los tests unitarios ejecutándose exitosamente

Riesgos Asociados:

- Limitaciones de rate limit de GitHub API
- Problemas de conectividad con Azure
- Errores en algoritmos de clasificación

Hito 3: Base de Datos Analítica Poblada (23/05/2025)

Entregables:

- Dataset completo con análisis de 400+ repositorios
- Base de datos con información normalizada y validada
- Métricas estadísticas preliminares calculadas
- Informe de análisis de datos con hallazgos principales
- Documentación de calidad de datos y metodología

Criterios de Aceptación:

- Análisis completo de al menos 400 repositorios estudiantiles

- Identificación de top 10 tecnologías por categoría
- Datos validados con precisión $\geq 95\%$
- Métricas de distribución tecnológica por año académico
- Detección de patrones significativos en uso de herramientas

Riesgos Asociados:

- Calidad insuficiente de datos en repositorios
- Sesgos en la muestra de proyectos analizados
- Problemas de rendimiento en procesamiento masivo

Hito 4: Dashboards Interactivos Implementados (20/06/2025)

Entregables:

- Dashboards Power BI completamente funcionales
- Visualizaciones interactivas con filtros dinámicos
- Reportes automatizados para diferentes audiencias
- Manual de usuario para administradores
- Sistema de actualización automática de datos

Criterios de Aceptación:

- Dashboards accesibles desde navegadores web modernos
- Tiempo de carga de visualizaciones < 5 segundos
- Filtros funcionales por curso, año, tecnología
- Exportación de reportes a PDF y Excel
- Satisfacción de usuarios $\geq 85\%$ en pruebas de usabilidad

Riesgos Asociados:

- Problemas de rendimiento en Power BI
- Dificultades de integración con fuentes de datos
- Resistencia al cambio por parte de usuarios

Hito 5: Proyecto Finalizado y Entregado (11/07/2025)

Entregables:

- Sistema completo funcionando en producción
- Documentación técnica y de usuario completa
- Informe final con conclusiones y recomendaciones
- Presentación ejecutiva para autoridades
- Plan de mantenimiento y sostenibilidad
- Código fuente documentado y versionado

Criterios de Aceptación:

- Sistema operativo al 100% de funcionalidad requerida
- Todos los requerimientos funcionales implementados

- Documentación completa y actualizada
- Capacitación realizada a usuarios finales
- Aprobación formal del proyecto por stakeholders
- Cumplimiento de presupuesto y cronograma

Riesgos Asociados:

- Problemas de última hora en integración
- Resistencia institucional para adopción
- Falta de recursos para mantenimiento futuro

Gestión de Hitos:

Seguimiento:

- Reunión semanal de avance hacia cada hito
- Métricas de progreso actualizadas diariamente
- Alertas tempranas por posibles retrasos
- Escalación automática de problemas críticos

Validación:

- Checklist de criterios de aceptación para cada hito
- Validación formal por stakeholders designados
- Documentación de aprobación o rechazo
- Plan de acciones correctivas si es necesario

II. Presupuesto

1. Planteamiento de aplicación del presupuesto

El presupuesto del proyecto se estructura considerando la naturaleza académica del mismo y la disponibilidad de recursos de la Universidad Privada de Tacna. Se priorizan herramientas de código abierto para minimizar costos de licenciamiento, mientras se aseguran recursos suficientes para infraestructura cloud y materiales necesarios.

La distribución presupuestaria se enfoca en tres áreas principales: costos operativos esenciales, infraestructura tecnológica y compensación por dedicación del equipo de desarrollo. Se han considerado contingencias del 10% para imprevistos técnicos y operativos.

2. Presupuesto

2.1 Costos Generales

Descripción	Cantidad	Costo Unitario (S/)	Total (S/)
Material de oficina	1 set	200	200

Impresiones y papelería	500 hojas	0.30	150
Licencias de software	-	0 (open-source)	0
Subtotal			350

2.2 Costos Operativos

Descripción	Duración	Costo Mensual (S/)	Total (S/)
Internet de alta velocidad	5 meses	60	300
Electricidad adicional	5 meses	0	0
Transporte y comunicaciones	5 meses	0	0
Subtotal			300

2.3 Costos de Infraestructura Cloud

Descripción	Duración	Costo Mensual (S/)	Total (S/)
Azure SQL Database (Basic)	5 meses	60	300
Azure Storage Account	5 meses	20	100
Power BI Pro licencias (2)	5 meses	40	200
Dominio web	1 año	100	100
Subtotal			700

2.4 Costos de Personal

Rol	Dedicación	Meses	Costo/Hora (S/)	Total (S/)
Desarrollador Principal (Angel)	25 hrs/sem	5	15	1,875
Desarrollador Backend (Piero)	25 hrs/sem	5	15	1,875
Supervisor Académico	4 hrs/sem	5	25	500
Soporte TI	3 hrs/sem	5	20	300
Subtotal				4,550

2.5 Resumen Total del Presupuesto

Categoría	Monto (S/)	Porcentaje
Costos generales	350	6.1%
Costos operativos	300	5.2%
Infraestructura cloud	700	12.2%

Costos de personal	4,550	79.1%
Subtotal	5,900	102.6%
Contingencia (10%)	-650	-11.3%
TOTAL PROYECTO	5,250	100%

3. Análisis de Factibilidad

3.1 Factibilidad Económica

El proyecto presenta una excelente relación costo-beneficio considerando:

- **Retorno de Inversión:** Se estima un ROI del 200% en el primer año mediante optimización de recursos tecnológicos y mejora en la calidad educativa.
- **Costos Evitados:** Reducción de S/2,000 anuales en análisis manuales y consultorías externas.
- **Beneficios Cuantificables:** Ahorro de 100 horas/año en tareas de análisis tecnológico para docentes.

3.2 Factibilidad Técnica

- **Recursos Disponibles:** La UPT cuenta con infraestructura suficiente para soportar el desarrollo y operación del sistema.
- **Tecnologías Probadas:** Uso de tecnologías maduras y estables (Python, Power BI, Azure SQL).
- **Competencias del Equipo:** El equipo de desarrollo posee las habilidades técnicas necesarias.

3.3 Factibilidad Operativa

- **Alineación Estratégica:** El proyecto se alinea con los objetivos de modernización tecnológica de la UPT.
- **Apoyo Institucional:** Respaldo confirmado de autoridades académicas y departamento de TI.
- **Usuarios Comprometidos:** Stakeholders académicos han expresado interés y compromiso con el proyecto.

4. Evaluación Financiera

4.1 Análisis de Flujo de Caja

Mes	Costos Fijos (S/)	Costos Variables (S/)	Total Mes (S/)	Acumulado (S/)
1	1,140	200	1,340	1,340
2	1,140	0	1,140	2,480

3	1,140	0	1,140	3,620
4	1,140	0	1,140	4,760
5	1,140	350	1,490	6,250
Ajuste			-1,000	5,250

4.2 Análisis de Sensibilidad

Escenario Optimista (15% reducción de costos):

- Uso de recursos universitarios adicionales
- Colaboración voluntaria de estudiantes avanzados
- **Costo Total:** S/4,463

Escenario Pesimista (20% incremento de costos):

- Necesidad de consultorías externas
- Retrasos que extiendan el cronograma
- **Costo Total:** S/6,300

4.3 Indicadores Financieros

- **Costo por Beneficiario:** S/13.13 por estudiante (400 estudiantes directos)
- **Costo por Tecnología Analizada:** S/105 por herramienta tecnológica (50 tecnologías)
- **Costo por Repositorio:** S/13.13 por proyecto analizado (400 repositorios)

4.4 Sostenibilidad Financiera

- **Mantenimiento Anual:** S/1,200 (infraestructura cloud + 40 horas de soporte técnico)
- **Actualizaciones Anuales:** S/800 (mejoras y nuevas funcionalidades)
- **Total Operación Anual:** S/2,000 (38% del costo inicial de desarrollo)

Anexo 01 – Requerimientos del Sistema "Análisis de Herramientas Tecnológicas UPT"

A1.1 Requerimientos Funcionales Detallados

RF001 - Extracción de Metadatos de Repositorios

- **Descripción:** El sistema debe conectarse a la API de GitHub para extraer metadatos de repositorios estudiantiles
- **Entrada:** URL de repositorio GitHub o lista de repositorios
- **Proceso:** Autenticación, consulta API, extracción de metadatos (nombre, descripción, lenguajes, fechas, colaboradores)
- **Salida:** Dataset estructurado con metadatos de repositorios
- **Criterios de Aceptación:**

- Procesar mínimo 400 repositorios sin errores
- Tiempo de procesamiento ≤ 30 segundos por repositorio
- Manejo de errores y reintentos automáticos
- **Prioridad:** Crítica

RF002 - Clasificación de Lenguajes de Programación

- **Descripción:** El sistema debe identificar y clasificar lenguajes de programación utilizados en cada proyecto
- **Entrada:** Archivos de código fuente y configuración del repositorio
- **Proceso:** Análisis de extensiones, contenido de archivos, manifiestos (package.json, pom.xml, requirements.txt)
- **Salida:** Lista de lenguajes con porcentaje de uso por proyecto
- **Criterios de Aceptación:**
 - Identificar correctamente los 15 lenguajes más comunes
 - Precisión $\geq 95\%$ en clasificación
 - Determinar lenguaje principal por proyecto
- **Prioridad:** Crítica

RF003 - Detección de Frameworks

- **Descripción:** El sistema debe identificar frameworks utilizados en los proyectos mediante análisis de código y documentación
- **Entrada:** Código fuente, archivos README, manifiestos de dependencias
- **Proceso:** Búsqueda de patrones, análisis de importaciones, lectura de archivos de configuración
- **Salida:** Lista de frameworks detectados por proyecto
- **Criterios de Aceptación:**
 - Detectar frameworks de al menos 5 lenguajes diferentes
 - Identificar correctamente frameworks populares (React, Django, Spring, etc.)
 - Distinguir entre frameworks frontend y backend
- **Prioridad:** Alta

RF004 - Identificación de Librerías

- **Descripción:** El sistema debe detectar librerías y bibliotecas utilizadas, clasificándolas por contexto de lenguaje
- **Entrada:** Archivos de dependencias, código fuente, documentación
- **Proceso:** Parsing de manifiestos, análisis de importaciones, búsqueda en documentación
- **Salida:** Lista de librerías categorizadas por lenguaje de programación
- **Criterios de Aceptación:**
 - Identificar librerías de Python, JavaScript, Java, C#
 - Clasificar por categoría (análisis de datos, interfaz, testing, etc.)
 - Detectar versiones cuando sea posible
- **Prioridad:** Media

RF005 - Detección de Bases de Datos

- **Descripción:** El sistema debe identificar una base de datos principal por proyecto, priorizando por relevancia
- **Entrada:** Código fuente, archivos de configuración, documentación README
- **Proceso:** Búsqueda de patrones de conexión, análisis de configuraciones, priorización de tecnologías
- **Salida:** Una base de datos principal identificada por proyecto
- **Criterios de Aceptación:**
 - Identificar solo una BD por proyecto (no duplicados)
 - Priorizar BD relacionales sobre NoSQL cuando haya ambas
 - Detectar PostgreSQL, MySQL, MongoDB, SQLite, SQL Server
- **Prioridad:** Alta

RF006 - Identificación de Herramientas CI/CD

- **Descripción:** El sistema debe detectar herramientas de integración y despliegue continuo implementadas
- **Entrada:** Archivos de configuración (.github/workflows, .gitlab-ci.yml, etc.), documentación
- **Proceso:** Análisis de archivos YAML/JSON de CI/CD, búsqueda de patrones en documentación
- **Salida:** Lista de herramientas CI/CD por proyecto
- **Criterios de Aceptación:**
 - Detectar GitHub Actions, GitLab CI, Jenkins, Travis CI
 - Identificar Docker y Kubernetes cuando estén presentes
 - Distinguir entre herramientas de CI y CD
- **Prioridad:** Media

RF007 - Generación de Reportes Estadísticos

- **Descripción:** El sistema debe generar reportes visuales con estadísticas de uso de tecnologías
- **Entrada:** Base de datos analítica poblada
- **Proceso:** Cálculo de métricas, agregaciones por categorías, generación de gráficos
- **Salida:** Dashboards interactivos en Power BI
- **Criterios de Aceptación:**
 - Reportes por año académico, curso, tecnología
 - Gráficos de distribución, tendencias temporales, comparaciones
 - Exportación a PDF y Excel
- **Prioridad:** Alta

RF008 - Sistema de Filtros Dinámicos

- **Descripción:** El sistema debe permitir filtrar y segmentar datos por múltiples criterios
- **Entrada:** Selecciones de usuario en interfaz web
- **Proceso:** Aplicación de filtros en tiempo real, actualización de visualizaciones
- **Salida:** Dashboards actualizados según filtros seleccionados
- **Criterios de Aceptación:**
 - Filtros por curso (SI783, SI784, SI885, etc.)
 - Filtros por año académico (2023, 2024, 2025)

- Filtros por tipo de tecnología (lenguajes, frameworks, BD)
 - Respuesta ≤ 3 segundos
- **Prioridad:** Media

A1.2 Requerimientos No Funcionales Detallados

RNF001 - Rendimiento

- **Descripción:** El sistema debe mantener tiempos de respuesta óptimos bajo cargas normales de trabajo
- **Especificaciones:**
 - Procesamiento: ≤ 30 segundos por repositorio analizado
 - Dashboards: Tiempo de carga ≤ 5 segundos
 - Reportes: Generación ≤ 2 minutos para reportes completos
 - Concurrencia: Soporte para 10 usuarios simultáneos
- **Método de Verificación:** Pruebas de carga automatizadas

RNF002 - Disponibilidad

- **Descripción:** El sistema debe estar disponible durante horarios laborales
- **Especificaciones:**
 - Disponibilidad: 99.5% durante horario laboral (8:00-18:00)
 - Tiempo de inactividad planificado: ≤ 2 horas/mes
 - Recuperación ante fallos: ≤ 15 minutos
- **Método de Verificación:** Monitoreo automático con alertas

RNF003 - Escalabilidad

- **Descripción:** El sistema debe poder crecer con el aumento de datos y usuarios
- **Especificaciones:**
 - Capacidad: Hasta 1000 repositorios analizados
 - Usuarios: Hasta 50 usuarios concurrentes
 - Datos: Base de datos hasta 10GB
 - Crecimiento: 20% anual en capacidad
- **Método de Verificación:** Pruebas de estrés graduales

RNF004 - Usabilidad

- **Descripción:** El sistema debe ser intuitivo y fácil de usar para usuarios no técnicos
- **Especificaciones:**
 - Curva de aprendizaje: ≤ 30 minutos para usuarios básicos
 - Satisfacción: $\geq 85\%$ en encuestas de usabilidad
 - Errores de usuario: $\leq 5\%$ en tareas comunes
 - Accesibilidad: Cumplimiento WCAG 2.1 nivel AA
- **Método de Verificación:** Pruebas con usuarios reales

RNF005 - Seguridad

- **Descripción:** El sistema debe proteger datos y accesos no autorizados

- **Especificaciones:**
 - Autenticación: Integración con Active Directory UPT
 - Encriptación: AES-256 para datos sensibles
 - Auditoría: Logs de acceso y modificaciones
 - Backup: Respallos diarios automáticos
- **Método de Verificación:** Auditoría de seguridad externa

RNF006 - Compatibilidad

- **Descripción:** El sistema debe funcionar en múltiples plataformas y navegadores
- **Especificaciones:**
 - Navegadores: Chrome 100+, Edge 100+, Firefox 100+
 - Dispositivos: Desktop, tablet (resolución mínima 1024x768)
 - Sistemas: Windows 10+, macOS 10.15+
 - APIs: GitHub API v4, Power BI REST API
- **Método de Verificación:** Pruebas de compatibilidad multi-plataforma

RNF007 - Mantenibilidad

- **Descripción:** El sistema debe ser fácil de mantener y actualizar
- **Especificaciones:**
 - Documentación: 100% de código documentado
 - Modularidad: Arquitectura basada en componentes independientes
 - Versionado: Control de versiones con Git
 - Logs: Sistema de logging detallado para debugging
- **Método de Verificación:** Revisión de código y documentación

A1.3 Reglas de Negocio

RN001 - Una Base de Datos por Proyecto

- Solo se puede asignar una base de datos principal por proyecto analizado
- Prioridad: PostgreSQL > MySQL > MongoDB > SQLite > Otros

RN002 - Anonimización de Datos

- Los datos personales de estudiantes deben ser anonimizados
- Solo se almacenan identificadores de usuario de GitHub (públicos)

RN003 - Cursos Válidos

- Solo se analizan proyectos de cursos autorizados: SI783, SI784, SI685, SI885, SI8811, SI888, SI889, SI982

RN004 - Retención de Datos

- Los datos se conservan por 2 años académicos
- Después se archivan o eliminan según políticas universitarias

RN005 - Validación de Repositorios

- Solo se procesan repositorios con README mínimo
- Se excluyen repositorios fork sin modificaciones significativas

A1.4 Casos de Uso Principales

CU001 - Analizar Repositorio

- **Actor:** Sistema automatizado
- **Precondición:** Repositorio GitHub válido y accesible
- **Flujo Principal:**
 1. Sistema obtiene URL del repositorio
 2. Autentica con GitHub API
 3. Extrae metadatos del repositorio
 4. Analiza archivos de código fuente
 5. Clasifica tecnologías utilizadas
 6. Almacena resultados en base de datos
- **Postcondición:** Repositorio procesado y tecnologías clasificadas

CU002 - Generar Dashboard

- **Actor:** Usuario académico (docente/administrador)
- **Precondición:** Usuario autenticado en el sistema
- **Flujo Principal:**
 1. Usuario accede al sistema web
 2. Selecciona filtros deseados (curso, año, tecnología)
 3. Sistema aplica filtros a los datos
 4. Genera visualizaciones actualizadas
 5. Muestra dashboard interactivo
- **Postcondición:** Dashboard mostrado con datos filtrados

CU003 - Exportar Reporte

- **Actor:** Usuario académico
- **Precondición:** Dashboard generado con datos válidos
- **Flujo Principal:**
 1. Usuario selecciona opción de exportar
 2. Elige formato de exportación (PDF/Excel)
 3. Sistema genera archivo con datos actuales
 4. Proporciona enlace de descarga
- **Postcondición:** Reporte exportado y disponible para descarga