****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**Informe Final**

**Dashboard de Monitoreo de Repositorios Académicos en GitHub: Tendencias en Desarrollo y Gestión de Proyectos de los estudiantes en la facultad de Ingeniería de Sistemas**

Curso: Inteligencia de Negocios

Docente: Patrick Cuadros Quiroga

Integrantes:

***Chambi Cori , Jerson Roni (2021072619)***

***Flores Quispe, Jaime Elias (2021070309)***

***Leyva Sardon, Elvis Ronald (2021072614)***

**Tacna – Perú**

***2025***

| CONTROL DE VERSIONES | | | | | |
| --- | --- | --- | --- | --- | --- |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

ÍNDICE GENERAL

[1. Antecedentes 4](#_tymzpzq3gzgo)

[2. Planteamiento del Problema 4](#_6z3cxxvsj6dj)

[3. Objetivos 5](#_fm5rjfyca7my)

[4. Marco Teórico 6](#_lke6yhp5nois)

[5. Desarrollo de la Solución 8](#_ax1aqvbs1x8a)

[6. Cronograma 19](#_9tjb6achfjrb)

[7. Presupuesto 20](#_27wei7ahhx3k)

[8. Conclusiones 21](#_b3gj94w9of2b)

[9. Recomendaciones 21](#_ad1iyu5vcmk1)

[10. Bibliografía 22](#_v0xlua9z4l8m)

### Antecedentes

El monitoreo y análisis de la actividad académica de los estudiantes es fundamental para la mejora continua en las instituciones educativas. En el ámbito de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Privada de Tacna, la evaluación de proyectos académicos a través de plataformas como GitHub ha sido realizada principalmente de manera manual, lo que lleva a una carga de trabajo elevada para los docentes y puede generar evaluaciones subjetivas. Además, la falta de métricas objetivas para medir el desempeño de los estudiantes dificulta la retroalimentación precisa y en tiempo real. En este contexto, el uso de herramientas automatizadas para la evaluación y seguimiento de proyectos académicos representa una solución viable y moderna.

### Planteamiento del Problema

1. Problema

El principal problema que enfrenta la Escuela Profesional de Ingeniería de Sistemas de la Universidad Privada de Tacna es la evaluación manual y subjetiva de los repositorios académicos de los estudiantes en GitHub. Este proceso es ineficiente, propenso a errores y no permite una medición objetiva y estandarizada de la calidad del trabajo de los estudiantes. La ausencia de un sistema automatizado que brinde métricas claras sobre la actividad de los repositorios, como la frecuencia de commits, el uso de ramas o la calidad del código, dificulta la retroalimentación oportuna y la mejora continua en el proceso de aprendizaje.

1. Justificación

El Dashboard de Monitoreo de Repositorios Académicos en GitHub es una solución innovadora que permite automatizar la evaluación de los proyectos de los estudiantes, proporcionando métricas claras y objetivas sobre la actividad en GitHub. Esta herramienta permitirá a los docentes ahorrar tiempo en el proceso de revisión y evaluación, mejorando la transparencia y objetividad en las calificaciones. Además, proporcionará a los estudiantes retroalimentación en tiempo real, permitiéndoles mejorar sus proyectos de manera continua. La implementación de este sistema contribuirá a la mejora de la calidad educativa en la Escuela Profesional de Ingeniería de Sistemas y a la modernización de los procesos académicos mediante el uso de herramientas tecnológicas avanzadas.

1. Alcance

El proyecto se enfoca en el desarrollo de un Dashboard de Monitoreo de Repositorios Académicos en GitHub para los estudiantes de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Privada de Tacna. El sistema se limitará a la evaluación y análisis de repositorios en GitHub, y no abarcará otras plataformas como GitLab o Bitbucket. El dashboard proporcionará métricas detalladas sobre la actividad de los repositorios (commits, pull requests, issues, uso de ramas) y facilitará la visualización de datos para los docentes y estudiantes. El sistema no incluirá la gestión de proyectos fuera de GitHub ni se extenderá a otras herramientas o sistemas educativos ajenos a GitHub.

### Objetivos

1. Objetivo general

Desarrollar un sistema automatizado para la evaluación de repositorios académicos en GitHub, utilizando un dashboard en Power BI que proporcione métricas objetivas y en tiempo real sobre la actividad de los proyectos estudiantiles en la Escuela Profesional de Ingeniería de Sistemas de la Universidad Privada de Tacna.

1. Objetivos específicos

* Automatizar la recolección y análisis de datos de los repositorios académicos de los estudiantes en GitHub (commits, pull requests, issues y tecnologías utilizadas) mediante la API de GitHub.
* Generar métricas de calidad que midan aspectos como la frecuencia de contribuciones, la complejidad del código, el uso de buenas prácticas de desarrollo, y la documentación de los proyectos.
* Desarrollar un dashboard interactivo en Power BI que permita a los docentes y estudiantes visualizar el desempeño de los proyectos de manera clara y objetiva.
* Reducir el tiempo de evaluación de los repositorios académicos en un 50%, optimizando el proceso de revisión y retroalimentación de los docentes.
* Fomentar las buenas prácticas de desarrollo (como el uso de issues, pull requests, y documentación) entre los estudiantes, alineando su trabajo con estándares profesionales.

### Marco Teórico

1. GitHub como Herramienta para el Desarrollo Académico

GitHub es una plataforma de desarrollo colaborativo basada en la web que permite gestionar repositorios de código fuente y fomentar la colaboración entre programadores y estudiantes. En el ámbito académico, GitHub es ampliamente utilizado por los estudiantes de ingeniería para gestionar sus proyectos y colaborar de manera eficiente. Al ser una herramienta de control de versiones basada en Git, GitHub permite un seguimiento detallado de los cambios realizados en los proyectos, lo cual es fundamental para la evaluación académica.

GitHub no solo facilita la gestión de código, sino que también ofrece funcionalidades clave como commits, pull requests, issues, y branches, que pueden ser aprovechadas para medir la actividad de los estudiantes en proyectos académicos, las cuales son indicadores clave de su progreso.

1. Evaluación de Proyectos Académicos en GitHub

La evaluación de proyectos en GitHub presenta un conjunto de desafíos. Tradicionalmente, los docentes deben realizar una revisión manual de los repositorios, lo que es un proceso largo y subjetivo. Además, la falta de métricas objetivas dificulta la evaluación precisa de la calidad del trabajo y la colaboración de los estudiantes. Esto genera una falta de transparencia y puede afectar la calidad de la retroalimentación.

El uso de herramientas automatizadas que extraigan métricas de las actividades de los estudiantes en GitHub podría mejorar esta situación, proporcionando a los docentes indicadores claros y cuantificables que reflejan el desempeño de los estudiantes.

1. Métricas en GitHub

* Frecuencia de Commits: La cantidad de cambios registrados en un repositorio. Una mayor frecuencia de commits puede reflejar un esfuerzo continuo y un compromiso con el proyecto.
* Uso de Pull Requests (PRs): La cantidad y calidad de los pull requests realizados, lo que indica la colaboración y la revisión del código por parte de otros miembros del equipo.
* Issues y Etiquetas: La utilización de issues para la gestión de tareas es una buena práctica que permite un seguimiento claro de las actividades realizadas y pendientes en el proyecto.
* Documentación: La presencia de archivos como README.md o informes técnicos que faciliten la comprensión del proyecto por otros desarrolladores o evaluadores.
* Complejidad del Código: La evaluación de la calidad del código mediante métricas como la complejidad ciclomática, que mide la cantidad de caminos independientes en el código fuente.

1. Power BI como Herramienta de Visualización

Power BI es una herramienta de visualización de datos de Microsoft que permite crear informes interactivos y dashboards. Su integración con GitHub API y la capacidad para importar y analizar datos de repositorios académicos hace que Power BI sea ideal para este proyecto, ya que puede proporcionar una representación visual de las métricas extraídas de los proyectos. Los docentes y estudiantes pueden visualizar los datos de manera clara, obteniendo insights relevantes sobre el progreso y la calidad del trabajo realizado.

1. Automatización de la Evaluación

La automatización de la evaluación de repositorios de GitHub no solo mejora la eficiencia del proceso, sino que también permite eliminar la subjetividad y mejorar la objetividad en la calificación. Al proporcionar métricas claras y análisis en tiempo real, los estudiantes pueden recibir retroalimentación constante sobre su desempeño, lo que les permite mejorar sus proyectos de manera continua.

1. Impacto de la Retroalimentación Temprana en el Rendimiento Académico

Varios estudios han demostrado que la retroalimentación temprana en el proceso de aprendizaje tiene un impacto positivo en el rendimiento académico de los estudiantes. En el caso de GitHub, las alertas sobre la actividad del repositorio y las métricas de desempeño pueden ayudar a los estudiantes a identificar áreas de mejora antes de la evaluación final del proyecto.

### Desarrollo de la Solución

1. Análisis de Factibilidad (técnico, económica, operativa, social, legal, ambiental)
   1. **Factibilidad Técnica**

**Hardware Disponible**

**Equipos de Desarrollo:**

Se necesitan equipos de desarrollo capaces de ejecutar software de desarrollo web y herramientas de análisis de datos. Los materiales previamente mencionados cumplen con las especificaciones mínimas requeridas:

* Procesador: Intel Core i5 12450H de 8 núcleos, útil para el manejo de tareas de programación, depuración y pruebas.
* Memoria RAM: De 8 a 16 GB de memoria DDR4 a 3200 mhz expandible.
* Almacenamiento: Disco sólido Nvme PCIe 4.0 de mínimo 1tb a 500gb para el sistema operativo. Esto asegura tiempos de carga cortos y un óptimo rendimiento general.
* Tarjeta Gráfica: Se usa la tarjeta gráfica integrada en el procesador Intel, pero también se puede hacer uso de una GPU dedicada como la Nvidia RTX 3060.

**Software**

**Aplicaciones y Herramientas de Desarrollo:**

* Visual Studio Code: Es el IDE principal para el desarrollo del proyecto, siendo compatible con los sistemas operativos Windows y macOS, ofreciendo extensiones que personalizan el entorno de trabajo.
* Terraform: Utilizado para la creación y gestión automatizada de la infraestructura en Azure, asegurando consistencia y escalabilidad.
* Power BI: Herramienta es para la creación de dashboards y visualizaciones de datos interactivos, además de ser poderosa para los análisis de datos.
* SQL Database: Base de datos transaccional utilizada para la gestión de datos académicos.

**Navegadores Web:** La plataforma debe ser compatible con los navegadores web más conocidos y utilizados tales como Google Chrome, Mozilla Firefox, Microsoft Edge, etc.

**Infraestructura en la Nube**

* **Grupo de Recursos:** Contenedor lógico, que organiza todos los recursos relacionados en la ubicación East US.
* **Azure SQL Server:** Servidor principal para alojar la base de datos SQL, donde las credenciales son gestionadas de forma segura (Versión utilizada: SQL Server 12.0.)
* **Azure SQL Database:** Base de datos con una capacidad de hasta 32 GB de almacenamiento, una capacidad mínima de 0.5 vCores para ahorrar costos cuando está inactiva y cuenta con una auto-pausa que se activa tras 60 minutos de inactividad.
* **Terraform:** Infraestructura definida como código para garantizar consistencia y despliegue automatizado de recursos en Azure.

**Automatización**

La creación y el despliegue de los recursos en Azure se gestionan mediante Terraform, lo que asegura que los entornos de desarrollo, pruebas y producción se configuren de forma uniforme.

**Infraestructura de Red**

* **Conexión a Internet:**

Una conexión a internet de alta velocidad es esencial para garantizar la disponibilidad y el acceso continuo a los servicios en la nube.

**Conclusión:** El proyecto cuenta con todos los recursos técnicos necesarios para su implementación exitosa.

* 1. **Factibilidad Económica**

**Costos Generales**

| **Material** | **Cantidad** | **Precio Unitario** | **Total** |
| --- | --- | --- | --- |
| Paquete de Hojas Bond | 1 | S/ 12.00 | S/ 12.00 |
| Lápices | 3 | S/ 1.60 | S/ 4.80 |
| **TOTAL** |  |  | **S/ 16.80** |

**Costos operativos durante el desarrollo**

Azure SQL Database (azurerm\_mssql\_database.db\_negocios\_u2)

| **Componente** | **Cantidad Mensual** | **Unidad** | **Costo Mensual** |
| --- | --- | --- | --- |
| Compute (serverless, GP\_S\_Gen5\_2) | Variable | vCore-hours | $0.52 por vCore-hora\* |
| Storage | 32 | GB | $3.68 |
| Long-term retention (LRS) | Variable | GB | $0.025 por GB\* |
| PITR backup storage (LRS) | Variable | GB | $0.10 por GB\* |

Storage Account Negocios U2 (azurerm\_storage\_account.negociosu2\_storage)

| **Componente** | **Cantidad Mensual** | **Unidad** | **Costo Mensual** |
| --- | --- | --- | --- |
| Capacity | Variable | GB | $0.0184 por GB\* |
| Write operations | Variable | 10k operations | $0.05 por 10k operaciones\* |
| List and create container operations | Variable | 10k operations | $0.05 por 10k operaciones\* |
| Read operations | Variable | 10k operations | $0.004 por 10k operaciones\* |
| All other operations | Variable | 10k operations | $0.004 por 10k operaciones\* |
| Blob index | Variable | 10k tags | $0.03 por 10k tags\* |

Storage Account - Terraform State (azurerm\_storage\_account.tfstate\_sa)

| **Componente** | **Cantidad Mensual** | **Unidad** | **Costo Mensual** |
| --- | --- | --- | --- |
| Capacity | Variable | GB | $0.0208 por GB\* |
| Write operations | Variable | 10k operations | $0.05 por 10k operaciones\* |
| List and create container operations | Variable | 10k operations | $0.05 por 10k operaciones\* |
| Read operations | Variable | 10k operations | $0.004 por 10k operaciones\* |
| All other operations | Variable | 10k operations | $0.004 por 10k operaciones\* |
| Blob index | Variable | 10k tags | $0.03 por 10k tags\* |

Resumen de Costos

| **Concepto** | Costo Mensual en $ | Costo Mensual en S/ |
| --- | --- | --- |
| **Costo Base** | $3.68 | S/13.31 |
| **Costo por Uso\*** | Variable | Variable |
| **TOTAL ESTIMADO** | $3.68 | S/13.31 |

**Costos del ambiente**

| CONCEPTO | DURACIÓN | COSTO EN $ | COSTO EN S/. |
| --- | --- | --- | --- |
| Azure SQL Database (azurerm\_mssql\_database.db\_negocios\_u2) | 12 meses | $44.16 | S/. 164.20 |
| Storage Account Negocios U2 (azurerm\_storage\_account.negociosu2\_storage) | 12 meses | $18.00 | S/. 66.96 |
| Storage Account Terraform State (azurerm\_storage\_account .tfstate\_sa) | 12 meses | $12.00 | S/. 44.64 |
| Licencia de PowerBl | 12 meses | Gratuito | Gratuito |
| **Total** |  | **$74.16** | **S/. 275.80** |

**Costos de personal**

| ROL | DURACIÓN DEL PROYECTO | COSTO POR MES | COSTO FINAL |
| --- | --- | --- | --- |
| DevOps | 3 meses | S/. 1200 | S/. 3600 |
| Analista de datos | 3 meses | S/. 1200 | S/. 3600 |
| Director de proyecto | 3 meses | S/. 1333.33 | S/. 4000 |
| **Total** |  |  | **S/. 11200** |

**Costos Totales del Proyecto (3 meses)**

| **Concepto** | **Costo en USD** | **Costo en S/.** |
| --- | --- | --- |
| Costos Generales | $4.52 | S/. 16.80 |
| Costos Operativos | $3.68 | S/. 13.31 |
| Costos de Ambiente | $74.16 | S/. 275.80 |
| Costos de Personal | $3,010.75 | S/. 11,200.00 |
| **TOTAL** | **$3,093.11** | **S/. 11,505.91** |

**Conclusión:** El proyecto es económicamente viable con una inversión total de $3,093.11, siendo el 97% del costo correspondiente al recurso humano.

* 1. **Factibilidad Operativa**

**Beneficios del Producto**

El Dashboard de Monitoreo de Repositorios Académicos en GitHub automatiza la evaluación de los proyectos de los estudiantes, reduciendo el tiempo de revisión manual para los docentes. Ofrece métricas objetivas sobre commits, calidad del código y documentación, garantizando evaluaciones más transparentes y consistentes. La integración con Power BI genera informes detallados que facilitan el análisis de tendencias y el rendimiento de los estudiantes, lo que mejora la eficiencia y toma de decisiones basada en datos.

**Impacto en los Usuarios**

Para los docentes, la automatización reduce la carga de trabajo y mejora la precisión en la retroalimentación. Los estudiantes reciben retroalimentación continua, lo que mejora la calidad de sus proyectos y fomenta el uso de buenas prácticas. Los administradores pueden identificar tendencias y áreas de mejora en el rendimiento estudiantil, optimizando la toma de decisiones académicas. La Escuela Profesional de Ingeniería de Sistemas se beneficia al modernizar la enseñanza con tecnologías disruptivas, mejorando la eficiencia y calidad educativa.

**Conclusión:** El sistema operará eficientemente, generando valor significativo para todos los stakeholders.

* 1. **Factibilidad Social**

La implementación del sistema de monitoreo de repositorios generará un impacto positivo en la comunidad académica, tanto para docentes como para estudiantes:

* **Mejora en la Evaluación Académica:** El sistema permitirá un análisis más preciso de la actividad de los estudiantes en sus repositorios, facilitando una evaluación objetiva y reduciendo el tiempo invertido en revisiones manuales.
* **Automatización de Procesos:** La generación de reportes y métricas reducirá la carga de trabajo de los docentes, optimizando su tiempo y mejorando la retroalimentación brindada a los estudiantes.
* **Facilidad de Uso y Accesibilidad:** Se diseñará una interfaz intuitiva para que tanto docentes como alumnos puedan utilizar el sistema sin necesidad de una capacitación extensa.
* **Fomento de Buenas Prácticas en Desarrollo de Software:** Al ofrecer métricas sobre la calidad del código y el uso de Git, el sistema ayudará a los estudiantes a mejorar su metodología de trabajo y fomentar el aprendizaje continuo.
* **Aumento de la Transparencia:** La posibilidad de monitorear la actividad en los repositorios permitirá detectar casos de deshonestidad académica, como el plagio, y promover una cultura de trabajo ético.
* **Impacto en la Institución:** La integración del sistema con plataformas académicas fortalecerá los procesos de enseñanza y posicionará a la institución como innovadora en el uso de tecnología para la educación.

**Conclusión:** El proyecto generará un impacto social positivo, beneficiando a toda la comunidad educativa.

* 1. **Factibilidad Legal**

**Cumplimiento con Regulaciones de Protección de Datos:**

El proyecto deberá cumplir con las normativas vigentes de protección de datos personales, como la Ley N° 29733, Ley de Protección de Datos Personales del Perú. Esta ley regula el tratamiento de datos personales para garantizar la privacidad de la información de los estudiantes. La plataforma deberá implementar medidas de seguridad que protejan los datos académicos y personales de los estudiantes para evitar cualquier tipo de vulnerabilidad o brecha de seguridad.

**Leyes de Propiedad Intelectual:**

La plataforma debe respetar las leyes de propiedad intelectual, asegurando que cualquier software, código o tecnología utilizada cuente con las licencias correspondientes. Esto incluye el uso de herramientas de software como Visual Studio Code y Power BI, las cuales tienen términos de uso que deben ser cumplidos.

**Normativas internas de la Universidad:**

Dado que el proyecto maneja información académica y administrativa de la Universidad Privada de Tacna, es crucial que todas las actividades de desarrollo y operación del sistema se alineen estrictamente con las normativas internas de la universidad. Esto incluye cumplir con las políticas de privacidad, seguridad de la información y cualquier otra regulación interna que rija el manejo y protección de los datos universitarios.

**Conclusión:** El proyecto cumple con todas las regulaciones legales aplicables, garantizando una operación segura y legal.

* 1. **Factibilidad Ambiental**

**Reducción del Uso de Papel:** Al digitalizar el monitoreo y análisis de repositorios en GitHub, se elimina la necesidad de imprimir informes o documentos físicos, contribuyendo a la conservación de recursos naturales y reduciendo la generación de residuos.

**Eficiencia Energética:** La implementación del sistema web optimiza el uso de recursos energéticos en comparación con los métodos tradicionales de revisión manual, ya que los procesos automatizados reducen el tiempo y esfuerzo requeridos.

**Impacto en la Huella de Carbono:** Al permitir que docentes y estudiantes accedan al sistema de forma remota, se minimiza la necesidad de desplazamientos físicos, reduciendo las emisiones de gases de efecto invernadero asociadas con el transporte.

**Gestión de Residuos:** La automatización de procesos y generación digital de reportes reduce el uso de materiales físicos, minimizando residuos innecesarios en el entorno académico.

**Cumplimiento de Normativas Ambientales:** El sistema cumple con principios de sostenibilidad al fomentar el uso responsable de la tecnología, reduciendo el impacto ambiental de las actividades académicas.

**Conciencia y Educación Ambiental:** La implementación del sistema fomenta el uso de tecnologías sostenibles, promoviendo la conciencia ambiental entre docentes y estudiantes y alentando la adopción de prácticas responsables en el ámbito académico.

**Conclusión:** El proyecto contribuye positivamente al medio ambiente mediante la digitalización y uso eficiente de recursos.

1. Tecnología de Desarrollo

| **Herramienta** | **Especificaciones** |
| --- | --- |
| IDE/Editor de Código | Visual Studio Code versión 1.100.3 |
| Base de Datos | Azure SQL database |
| Servidor | Azure SQL server |
| Almacenamiento de Archivos | Azure Blob Storage |
| CI/CD y Automatización | GitHub Actions |
| Business Intelligence | PowerBI |
| Lenguaje de Programación | Python 3.10 |
| Cliente de Base de Datos | Azure Data Studio |
| Infraestructura y gestión de recursos de azure | Terraform 1.9.8 |
| Estimación de costos de infraestructura | Infracost 0.10.38 |

1. Metodología de implementación (Documento de VISIÓN, SRS, SAD)

**Fases de implementación**

**Fase 1: Preparación y Configuración del Entorno**

* Creación del Grupo de Recursos:
  + Configuración del grupo de recursos "inteligencia-negocios" en la región East US.
* Configuración del Servidor SQL:
  + Implementación del servidor SQL "negocios-u2-sql.database.windows.net" con versión 12.0.
  + Configuración del usuario administrador (adminsql) y contraseña.
* Creación de la Base de Datos:
  + Base de datos "SoftRepoTrack" con:
    - Capacidad de almacenamiento: 32 GB.
    - Capacidad mínima: 0.5 vCores (para ahorro en inactividad).
    - Auto-pausa: 60 minutos.
* Infraestructura de Aplicación:

**Fase 2: Extracción, Transformación y Carga (ETL)**

* Extracción de Datos:
* Transformación de Datos:

**Fase 3: Visualización y Desarrollo de Dashboards**

* Conexión de Power BI:
  + Configuración de la conexión entre Power BI y la base de datos Azure SQL.
* Desarrollo de Dashboards:
  + Creación de visualizaciones interactivas:
  + Implementación de
  + Optimización de

**Fase 4: Pruebas y Validación**

* Pruebas de Funcionalidad:
* Pruebas de Rendimiento:

**Fase 5: Despliegue y Monitoreo**

* Publicación del Dashboard en Power BI Service:
  + Configuración de accesos y permisos a usuarios finales.
* Monitoreo del Desempeño:
  + Uso de herramientas de Azure para monitorear la base de datos y la aplicación.
  + Mantenimiento periódico y ajustes de optimización en Power BI.

### Cronograma

| **Fase** | **Tarea** | **Duración (días)** | **Inicio** | **Fin** |
| --- | --- | --- | --- | --- |
| Gestión de Proyectos | Inicio | 5 | 15/03/2025 | 19/03/2025 |
| Registro de Interesados | 3 | 20/03/2025 | 22/03/2025 |
| Plan de Proyecto | 2 | 25/03/2025 | 26/03/2025 |
| Informe de Estado del Proyecto | 2 | 27/03/2025 | 28/03/2025 |
| Reunión de Coordinación Semanal | 1 | 01/04/2025 | 01/04/2025 |
| Cierre del Proyecto | 4 | 07/06/2025 | 10/06/2025 |
| Fase de Concepción | Especificación de Requerimientos del Software | 4 | 02/04/2025 | 05/04/2025 |
| Documento Visión | 4 | 08/04/2025 | 11/04/2025 |
| Fase de Elaboración | Modelo de Casos de Uso | 3 | 14/04/2025 | 16/04/2025 |
| Diagrama de Clases | 3 | 17/04/2025 | 19/04/2025 |
| Documento de Arquitectura de Software | 4 | 22/04/2025 | 25/04/2025 |
| Fase de Construcción | Dashboard de Análisis de Métricas | 3 | 26/04/2025 | 28/04/2025 |
| Dashboard de Rendimiento Académico | 5 | 29/04/2025 | 03/05/2025 |
| Implementación de Filtros y Visualizaciones | 4 | 06/05/2025 | 09/05/2025 |
| Fase de Transición | Pruebas de Iteración Aprobadas | 3 | 12/05/2025 | 14/05/2025 |
| Manual de Usuario | 5 | 15/05/2025 | 19/05/2025 |
| Informe Final del Proyecto | 4 | 20/05/2025 | 23/05/2025 |
| Adquirir Tecnología | Configuración del Servidor | 8 | 26/05/2025 | 02/06/2025 |

### Presupuesto

| CONCEPTO | COSTO DÓLARES | COSTO SOLES |
| --- | --- | --- |
| Costos Generales | $4.52 | S/. 16.80 |
| Costos Operativos | $3.68 | S/. 13.31 |
| Costos de Ambiente | $74.16 | S/. 275.80 |
| Costos de Personal | $3,010.75 | S/. 11200 |
| **Total** | **$3,093.11** | **S/. 11,505.91** |

### Conclusiones

El proyecto Dashboard de Monitoreo de Repositorios Académicos en GitHub representa una solución innovadora para mejorar la evaluación académica de los estudiantes en la Escuela Profesional de Ingeniería de Sistemas de la Universidad Privada de Tacna. Al automatizar el proceso de revisión de repositorios en GitHub, el sistema no solo optimiza el tiempo de los docentes, sino que también ofrece una evaluación objetiva y basada en datos, lo que mejora la transparencia y precisión en las calificaciones. Además, al integrar herramientas como Power BI, proporciona a docentes y estudiantes un acceso fácil y visual a las métricas de desempeño, fomentando la mejora continua. Este sistema permite a los estudiantes recibir retroalimentación en tiempo real, lo que les brinda la oportunidad de mejorar sus proyectos antes de la entrega final. De esta manera, el proyecto no solo optimiza la gestión educativa, sino que también promueve el uso de buenas prácticas de desarrollo y fortalece la formación profesional de los estudiantes, alineándolos con los estándares tecnológicos actuales. En general, la implementación de este sistema contribuirá significativamente a la mejora de la calidad educativa, facilitando un entorno de aprendizaje más eficiente y enfocado en el desarrollo de habilidades prácticas.

### Recomendaciones

* Se recomienda implementar talleres periódicos para asegurar que tanto los docentes como los estudiantes aprovechen al máximo las funcionalidades del sistema, especialmente en cuanto a la gestión de pull requests, issues, y documentación en GitHub.
* En el futuro, sería beneficioso considerar la expansión del sistema para incluir plataformas adicionales como GitLab o Bitbucket, con el fin de cubrir un mayor número de repositorios utilizados por los estudiantes en otros cursos.
* Además de medir la actividad individual, es importante incorporar métricas que evalúen la colaboración entre los miembros de un equipo, como el número de PRs comentados o la resolución conjunta de issues, para evaluar habilidades blandas esenciales en el entorno profesional.
* Se recomienda integrar el sistema con plataformas de gestión educativa como Moodle o Google Classroom, para centralizar la información académica y vincular las métricas de GitHub con las calificaciones y retroalimentación de los cursos.
* Es crucial establecer un plan de mantenimiento para el sistema, asegurando que se actualicen las métricas y funcionalidades con el tiempo, para adaptarse a los cambios en las prácticas de desarrollo y las necesidades académicas. Además, se recomienda realizar encuestas periódicas a los usuarios para identificar áreas de mejora y optimizar la experiencia del usuario.

### Bibliografía

Natarajan, T., y Shanmugavadivu, P. (2024). Desarrollo basado en el comportamiento y marco de métricas para prácticas ágiles mejoradas en equipos Scrum. *Inf. Softw. Technol.* , 170, 107435. <https://doi.org/10.1016/j.infsof.2024.107435>

Jones, C. (2009). Mejores prácticas de ingeniería de software.. <https://doi.org/10.1002/9781119092919.ch15>

Patani, P., Tiwari, S. y Rathore, S. (2024). El impacto de GitHub en el aprendizaje y la participación de los estudiantes en un curso de ingeniería de software. *Comput. Appl. Eng. Educ.* , 32. <https://doi.org/10.1002/cae.22775>

Borges, H. y Valente, M. (2018). ¿Qué hay en una estrella de GitHub? Comprensión de las prácticas de asignación de estrellas a repositorios en una plataforma de programación social. *J. Syst. Softw.* , 146, 112-129. <https://doi.org/10.1016/j.jss.2018.09.016>