

Plan de Proyecto

Telito Developer Portal

Brenda Tumbalobos Cubas

Profesor del curso

TEL137 – Gestión de Servicios de TICS

Especialidad de Ingeniería de las Telecomunicaciones

Pontificia Universidad Católica del Perú

2025



Tarea Académica

Entrega: Fin del ciclo 2025-2

1 Título del Proyecto

Telito Developer Portal.

2 Coordinador del Proyecto

Brenda Tumbalobos Cubas

3 Clientes

- Jefes de Práctica y Profesor

4 Supervisores

Jefes de práctica del curso:

- Morillos, Jose Luis
- Roldán, Stefano
- Rengifo, Lewis

5 Objetivo del Proyecto

Diseñar e implementar un Developer Portal centralizado que permita a los distintos perfiles técnicos y de negocio (Desarrolladores, Product Owners, QA) acceder, consultar y gestionar de manera eficiente la documentación, contratos de APIs y recursos técnicos, así como interactuar con un asistente virtual (chatbot) que facilite la navegación del catálogo y la resolución de dudas

6 Alcance

El alcance del trabajo incluye las fases de planificación, ejecución e implementación del sistema. Cada cierre de fase requerirá la verificación del jefe de práctica y en caso lo requiera, la validación del profesor.

7 Presupuesto

50\$ en créditos de AWS.



8 Entregables

Se considerará lo siguiente:

- Sistema desplegado en la nube AWS.
- Repositorio GitHub
- Documentación del Proyecto
- Proyecto y tracking en Jira
- Flujo de usuario en Miro

9 Tiempo de Implementación Estimado

El proyecto tiene un tiempo estimado de 4 meses, empezando el 20 de agosto del 2025 y finalizando el 16 de diciembre del 2025 (tiempo final estimado).

10 Estrategia de control de versiones

Se utilizará github como herramienta para el control de versiones e integración del proyecto grupal.

11 Composición del equipo de trabajo

Serán 2 grupos de trabajo, 6 alumnos cada uno. Cada grupo tendrá un jefe de práctica, quien será el coach y encargado de realizar las coordinaciones a nombre del grupo con el profesor y los jefes de práctica del curso. El grupo deberá de tener un nombre para su participación.

12 Criterios evaluación

Criterios de evaluación	Puntaje
Nota de cada entregable <ul style="list-style-type: none">- Serán 12 notas de entregables.- A criterio del jefe de práctica.- Se verá impactado por las autoevaluaciones de todos los participantes.- Las autoevaluaciones serán promediadas antes del parcial y el restante para el examen final.- La Documentación Final es parte del entregable.	20

13 Características formales

- El plagio dará lugar a la descalificación del trabajo, con nota cero. La copia parcial o total del trabajo ya publicado en internet en alguna plataforma dará lugar a la descalificación.
- Consigne que todos los alumnos hayan realizado commits usando GIT como prueba de apoyo grupal y demostración sobre las notas de autoevaluación.
- Luego de cada corrección de entregable, en la siguiente entrega se deberá de presentar lo corregido y lo que se pide en la semana.

14 Instrucciones para la entrega

- Se deberá entregar cada entregable en la plataforma educativa Paideia PUCP antes de iniciar cada laboratorio, en caso contrario **se hará un descuento de 5 puntos por entrega fuera de horario.**
- En caso de subir el proyecto, subirlo en formato zip.
- En caso de tenerlo en la nube, colocar un archivo de texto con la ip.
- Para la presentación final, se deberá de presentar un documento en PDF y un archivo zip con la siguiente información:
 - Nombre del grupo
 - Integrantes
 - Nombre del Repositorio y URL
 - Roles y Credenciales - Usuarios de prueba
 - Arquitectura de la plataforma
 - Reporte de webservices (postman)
 - Análisis de costos y proyección de recuperación de la inversión.
 - Flujo de usuario por cada rol (colocar link del MIRO BOARD)
 - Link de videos de cómo crear usuarios o realizar los flujos principales.
 - Código fuente en zip.

15 Puntaje

Esta actividad será evaluada como promedio de los entregables y las tareas en clase acumuladas. Dicha nota representa el 15% de la nota final del curso. La presentación parcial del proyecto constituye la nota del examen parcial. La presentación final del proyecto tiene una nota aparte que constituye parte del examen final del curso. La documentación del proyecto, así como los avances del JIRA forman parte del examen parcial y final del curso.

16 Descripción del Proyecto

El proyecto consiste en la gestión de una plataforma web para reserva de canchas y servicios deportivos en general.

Deberá tener diferentes roles tanto internos como externos (usuarios finales) y permitir el pago a través de subir una imagen y que el usuario interno sea capaz de validarlo.

Actores

- Superadmin (deberá ser desarrollado por los alumnos a su propio criterio)
- Desarrolladores
- Product Owner
- QA

Flujos

- **Desarrollador**

1. Ingreso y Autenticación
 - El desarrollador ingresa al portal.



- Si es externo: onboarding sencillo con registro/authenticación segura (OAuth2, SSO).
- Si es interno: acceso con credenciales corporativas.
- 2. Exploración del Catálogo
 - Accede a un catálogo centralizado de APIs y recursos.
 - Puede filtrar/buscar APIs por dominio (ej. pagos, clientes, logística) o tags (REST, GraphQL, eventos, etc.).
 - Visualiza de inmediato:
 - Descripción funcional.
 - Contrato de API (OpenAPI/Swagger).
 - Versiones disponibles.
 - Estado (prod, QA, deprecated).
- 3. Revisión de Documentación
 - Accede a documentación con guías paso a paso.
 - Encuentra ejemplos de uso, snippets de código (curl, Java, Python, JS, etc.).
 - Lee políticas de seguridad, límites de rate-limiting y requisitos de autenticación.
- 4. Prueba de la API (Playground/Sandbox)
 - Directamente en el portal (o con Postman collections) prueba los endpoints con datos de prueba.
 - Validación rápida: request → response → status.
- 5. Soporte y Dudas
 - Si tiene preguntas:
 - Chatbot le da respuestas rápidas (FAQ, guías).
 - Si no se resuelve, el chatbot abre un ticket automáticamente a través de un formulario.
 - Puede ver el estado del ticket en el mismo portal.
- 6. Onboarding Técnico (Access Management)
 - Si necesita acceso real:
 - Solicita API Key o Client ID/Secret (workflow automático de aprobación).
 - El sistema valida permisos según su rol y proyecto.
- 7. Integración en el Proyecto
 - Descarga SDKs/librerías o usa snippets.
 - Integra la API en su código.
 - Valida con entornos de QA/Sandbox antes de pasar a PROD.
- 8. Deploy & Monitoreo
 - Una vez en PROD:
 - Puede ver métricas de consumo de su API (dashboards en el portal).
 - Alertas de errores o caídas notificadas vía email/portal.
- 9. Feedback y Mejora Continua
 - El portal permite dejar feedback (ej. ratear la documentación, sugerir mejoras).
 - Ese feedback va al backlog de evolución del portal/APIs.

• **Quality Assurance (QA)**

- 1. Ingreso y Autenticación
 - Acceso seguro con credenciales corporativas (SSO o LDAP).
 - Perfil detectado como QA → acceso a entornos de prueba, no a llaves de producción.
- 2. Exploración del Catálogo de APIs
 - Busca APIs que debe validar.
 - Ve especificaciones técnicas, contratos (OpenAPI), políticas de versionado y entornos disponibles (QA, Sandbox, PROD).
- 3. Revisión de Documentación de Testing
 - Encuentra guías de QA:
 - Casos de prueba recomendados.
 - Ejemplos de requests/responses esperados.
- 4. Acceso a Entornos de QA/Sandbox
 - Obtiene credenciales temporales o tokens limitados para pruebas.
 - Usa Playground del portal o colecciones de Postman preconfiguradas.

5. Ejecución de Pruebas
 - Valida endpoints, performance y errores.
 - Usa herramientas integradas de monitoreo/logs (ej. ver errores en responses).
6. Reporte de Incidencias
 - Si encuentra defectos:
 - Sube un issue desde el portal (chatbot/formulario).
 - Adjunta logs o capturas desde la misma interfaz.
7. Seguimiento de Tickets
 - Visualiza el estado de los issues en el portal.
 - Notificaciones automáticas de cambios (ej. bug corregido, retest requerido).
8. Cierre y Validación Final
 - QA aprueba la funcionalidad y marca como validado.
 - Feedback sobre la calidad de la documentación o facilidad de prueba.

• **Product Owner**

1. Ingreso y Autenticación
 - Acceso con su perfil de negocio/técnico.
 - Permisos para visualizar APIs, documentación de alto nivel y métricas de consumo.
2. Exploración del Catálogo
 - Busca APIs disponibles que puedan usarse en su producto.
 - Visualiza descripción funcional, casos de uso, KPIs asociados.
 - Puede identificar si la API está en producción, beta o deprecated.
3. Revisión de Documentación Funcional
 - Accede a documentación no técnica:
 - Casos de uso de negocio.
 - Ejemplos de integración en productos reales.
 - Beneficios y limitaciones de cada API.
4. Gestión de Acceso y Aprovisionamiento
 - Solicita acceso para su equipo (desarrolladores/QA).
 - Aprueba solicitudes de llaves de API o entornos según su rol.
5. Monitoreo y Métricas de Uso
 - Desde el portal puede ver:
 - Número de llamadas a la API.
 - Latencia promedio y errores.
 - Costos asociados (si aplica).
6. Comunicación y Feedback
 - Si el PO identifica una necesidad de mejora o nueva API:
 - Puede levantar un requerimiento de producto desde el portal.
 - Canal integrado al backlog del área de APIs.
7. Gestión de Roadmap
 - El portal ofrece visibilidad sobre el estado de evolución de APIs (nuevas, en desarrollo, próximas en roadmap).
 - PO puede alinear su producto a esas evoluciones.

Tener en cuenta que deben de expandirse en los requerimientos y dar un entregable para ser usado al cliente. Recordar que debe de ser lo más intuitivo para los vecinos y el personal interno.

17 Funcionalidades generales

- Todos los usuarios tienen sección de notificaciones. Con excepción del SuperAdmin que solo tiene cambio de contraseña.
- Todos los usuarios deberán de registrarse a través de una consulta con webservice de la RENIEC para validación de datos.
- El superadmin deberá poder loguearse como cualquier usuario sin necesidad de acceder a la clave personal del usuario.

18 Requerimientos no funcionales

- La aplicación debe ser responsive para celulares
- Proyectos y servicios web publicados en GitHub
- Proyecto desplegado en un proveedor de nube
- El proyecto debe usar servicios cloud de un proveedor para las funcionalidades que requieras procesamiento local tales como:
 - Envío de correo.
 - Almacenamiento de data, archivos e imágenes
 - Balanceo de carga
 - Chatbot y/o asistente virtual.
 - Y otros servicios que sean requerido por la plataforma.