|  |  |
| --- | --- |
| **NOMBRES ESTUDIANTES:** | Sebastián Donoso  Ismael Toala |
| **FECHA:** | 16-06-2025 |
|  |  |
|  |  |
| **TEMA:** | Proyecto 1 Bimestre |
|  |  |

**Nombre del Proyecto:** PoliCampus

**Introducción:**

PoliCampus es una plataforma web desarrollada para la Escuela Politécnica Nacional con el fin de mejorar la experiencia universitaria mediante el acceso centralizado a información clave. El sistema integra servicios como cafetería, transporte, asociaciones y un mercado estudiantil digital, facilitando la vida académica y extracurricular del estudiante.

**Objetivos:**

* Mejorar la comunicación institucional con los estudiantes.
* Optimizar el tiempo mediante acceso rápido a servicios universitarios.
* Fomentar la participación estudiantil y el sentido de comunidad.
* Facilitar el intercambio económico entre estudiantes mediante PoliMarket.

# 1. Entregables del Proyecto

## **1.1 Estrategia de Ramificación (Branching Strategy)**

**Tipo de estrategia:** GitFlow

GitFlow es una estrategia robusta para el versionamiento y organización del desarrollo colaborativo en proyectos que utilizan Git. Permite aislar el desarrollo de nuevas funcionalidades (feature/) respecto a la rama de integración (develop) y producción (main). También contempla ramas específicas para correcciones urgentes (hotfix/) o preparación de releases (release/).

**Estructura de ramas:**

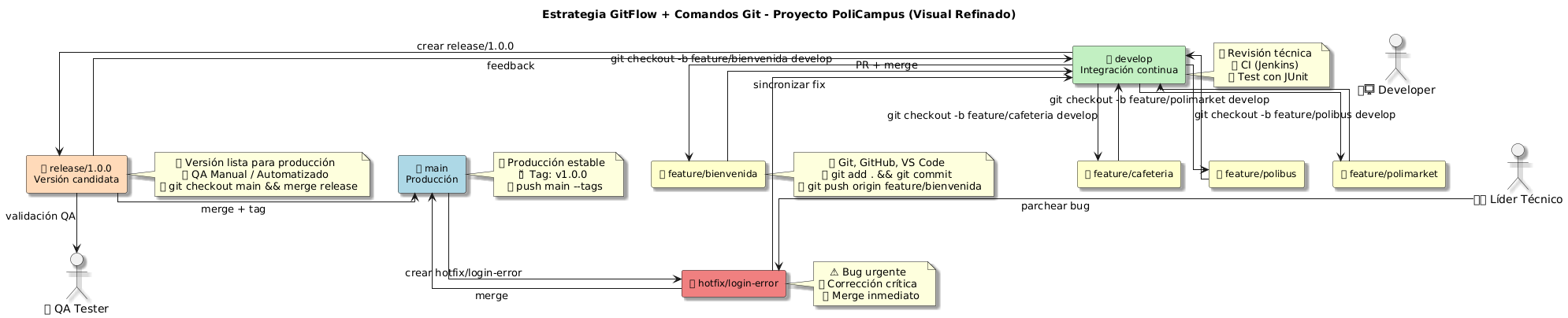
|  |  |
| --- | --- |
| **Rama** | **Propósito** |
| main | Código estable en producción |
| develop | Integración de nuevas funcionalidades |
| feature/\* | Desarrollo de funcionalidades (1 por HU) |
| release/\* | Preparación de versiones estables |
| hotfix/\* | Correcciones críticas en producción |
| bugfix/\* | Correcciones menores en desarrollo |

**Reglas de fusión (merge):**

* Todo feature/\* debe hacer un **Pull Request (PR)** hacia develop tras ser revisado y aprobado por al menos un miembro del equipo técnico.
* Al finalizar un sprint, se crea una rama release/x.y.z desde develop. Tras ajustes y pruebas, esta rama se fusiona en main y en develop.
* En caso de errores en main, se crea un hotfix/\*, se corrige el problema y se fusiona de forma urgente a main y develop.
* Toda rama fusionada debe ser eliminada para mantener un historial limpio (git branch -d nombre-rama).

**Roles y responsabilidades en GitFlow**

|  |  |
| --- | --- |
| **Rol** | **Responsabilidades clave en el flujo GitFlow** |
| **Developer** | Crea ramas feature/\*, hace commits, push y PR hacia develop. |
| **Líder Técnico** | Revisa PR, gestiona ramas release/\*, aprueba merge a main y etiqueta versiones. |
| **QA Tester** | Verifica versiones en release/\* antes de pasar a producción. |
| **DevOps/CI Engineer** | Automatiza el proceso: build, test, checkstyle y despliegue vía Jenkins y Docker. |

**Diagrama PlantUML del flujo GitFlow en PoliCampus:**

## **1.2 Plan de Mantenimiento**

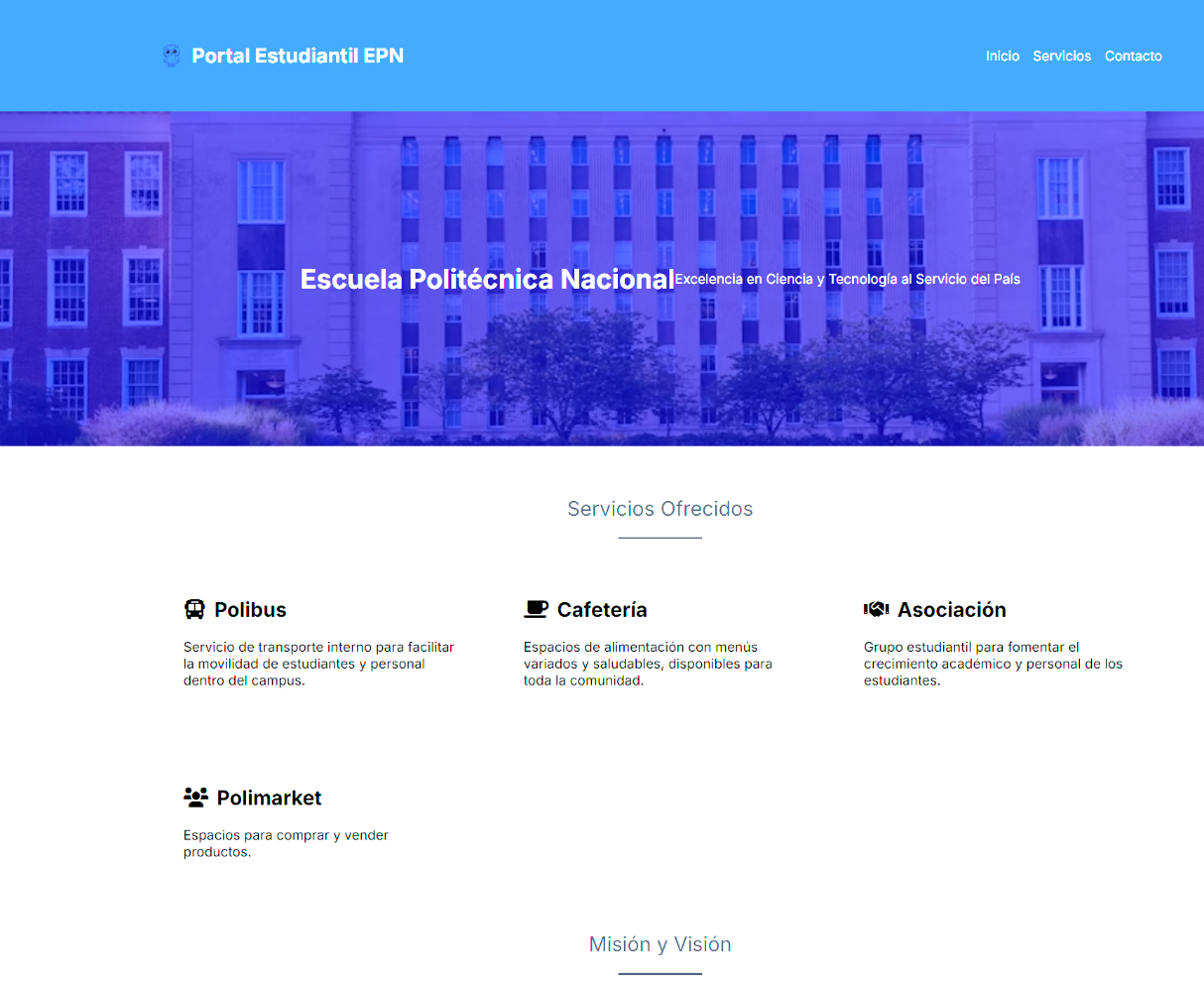
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo de Mantenimiento** | **Descripción** | **Funcionalidades Afectadas** | **Frecuencia Propuesta** | **Herramientas / Acciones** |
| **Correctivo** | Resolución de errores identificados por usuarios o QA. | Todas (bienvenida, cafetería, polibus, poliMarket) | Según reporte (prioridad alta en hotfix) | Issues GitHub, revisión de logs, pruebas JUnit específicas, rama hotfix/\*. |
| **Adaptativo** | Ajustes por cambios en dependencias o compatibilidad (Java, Maven, navegadores). | Todas, especialmente interfaz y backend. | Mensual o según actualización de dependencias | Actualización de pom.xml, pruebas de regresión, CI con Jenkins. |
| **Perfectivo** | Mejoras de rendimiento, experiencia de usuario (UX/UI), limpieza de código. | Cafetería (carga dinámica), PoliMarket (búsqueda y filtros). | Cada 2 sprints (quincenal) | Refactorización con SonarQube, optimización de consultas, lazy loading. |
| **Preventivo** | Acciones para evitar futuros errores: pruebas automáticas, refactorización, backup. | Backend (servicios Polibus y base de datos), controladores generales. | Final de cada sprint (semanal) | Aumento de cobertura de pruebas con JUnit, validaciones automáticas en Jenkins. |

## **1.3 Prototipo**

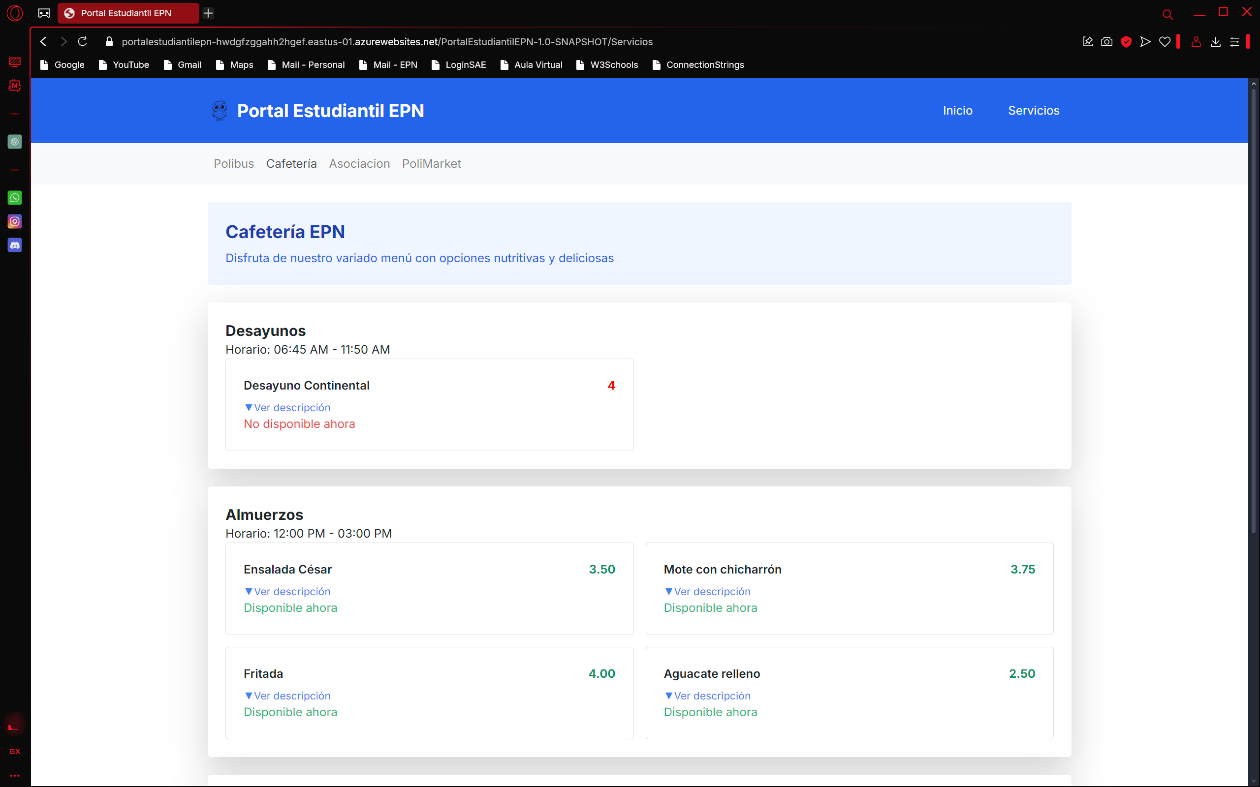
**Prototipo visual navegable:**

El prototipo fue elaborado en **Figma** e incluye las pantallas clave:

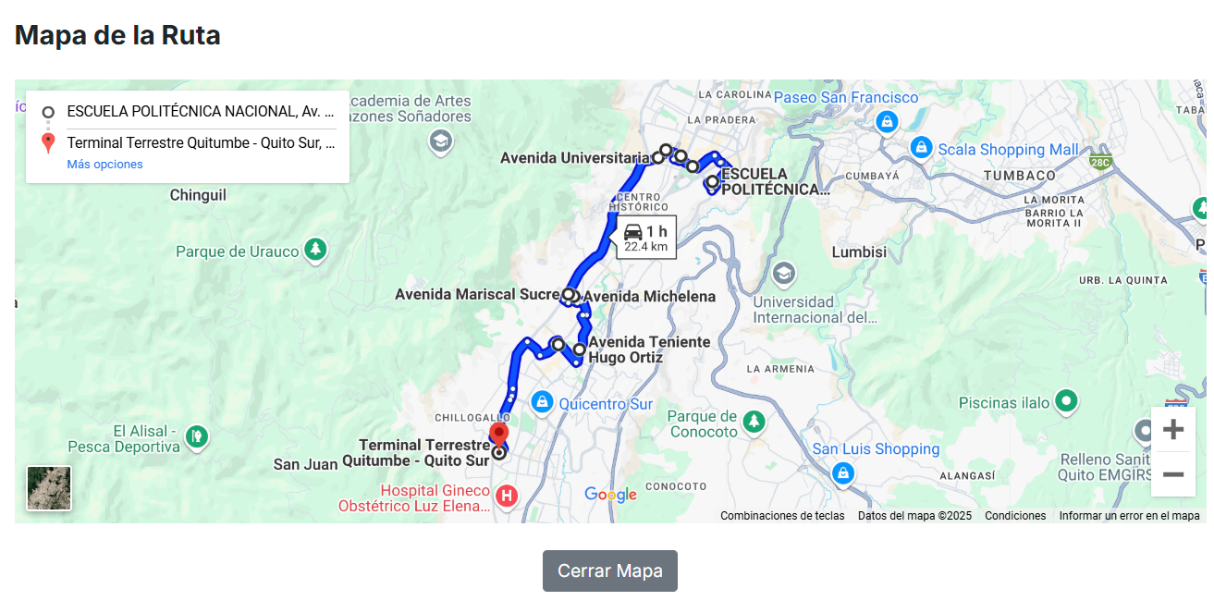
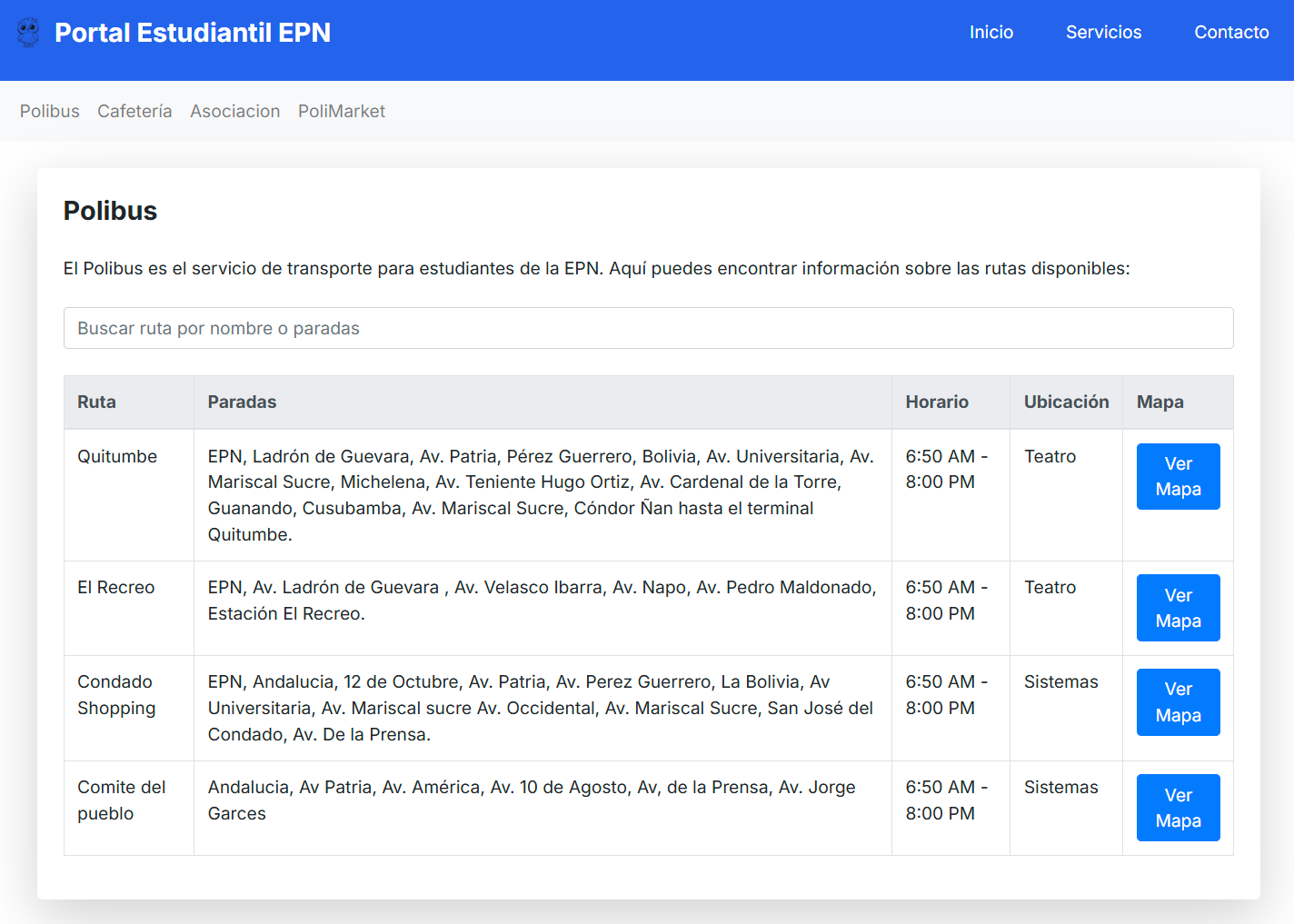
* Página de bienvenida



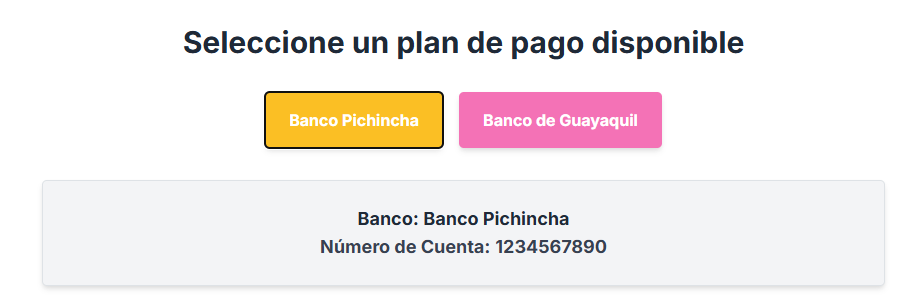
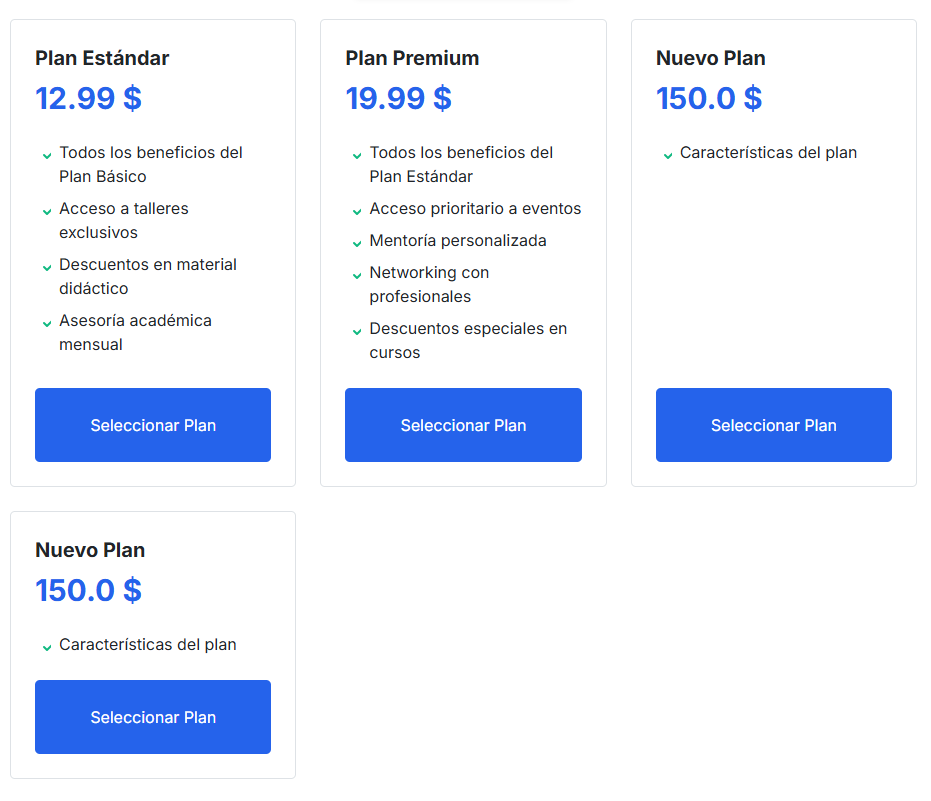
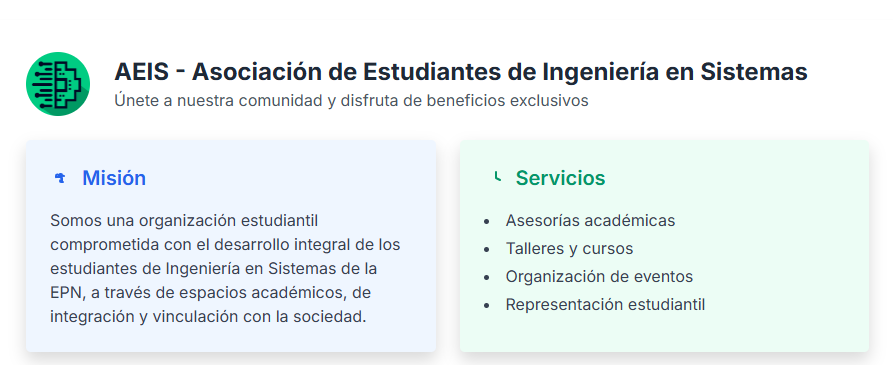
* Menú de cafeteria



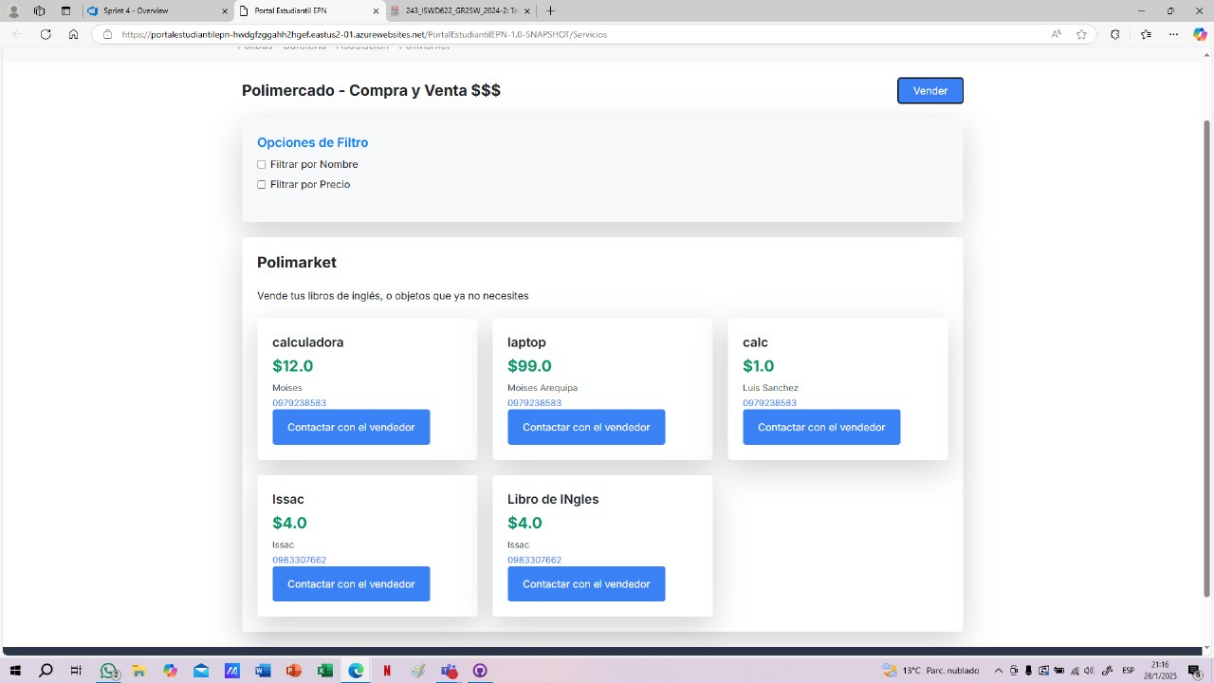
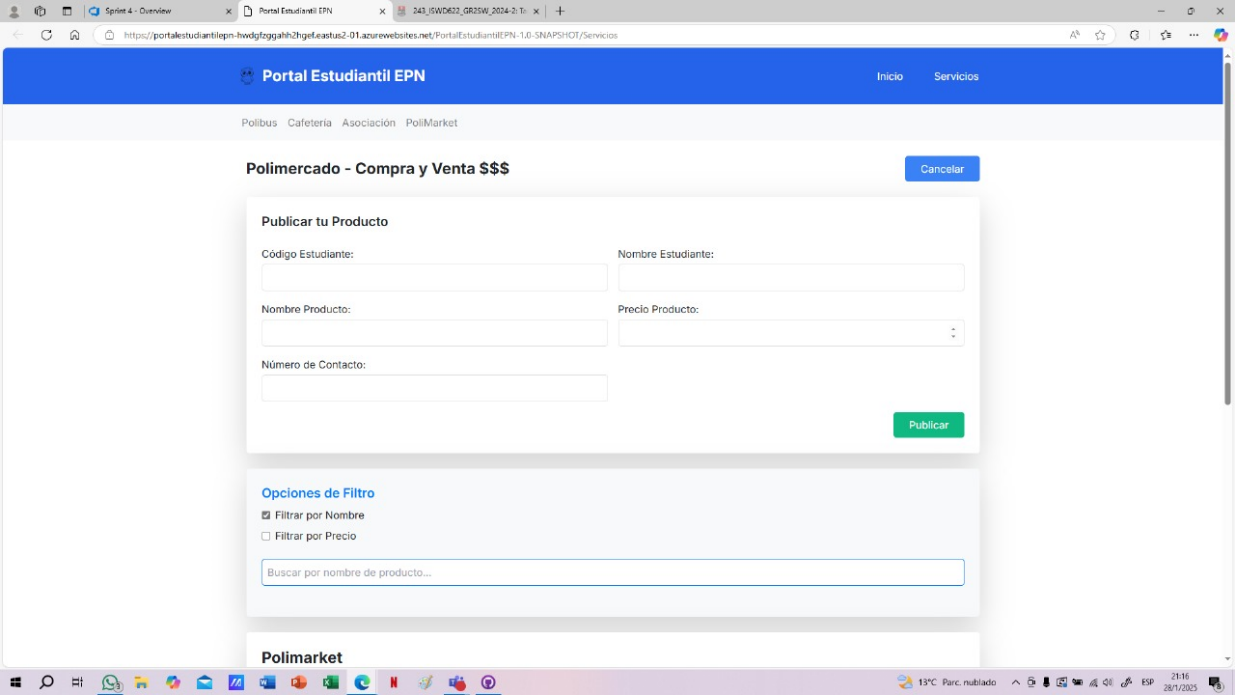
* Módulo de rutas Polibus



* Perfil y planes de asociaciones estudiantiles



* Interfaz de publicación y búsqueda de productos en PoliMarket



## **1.4 Historias de Usuario / Casos de Uso**

**Iteración 1 (Semana 1)**

**Objetivo Comercial de la Iteración:** Atraer a los estudiantes a la plataforma y generar un primer impacto positivo presentando una página de inicio atractiva y funcional, junto con la información esencial del servicio de cafetería. Se busca que los estudiantes puedan acceder fácilmente a los menús y precios, incentivando el uso de la plataforma desde su lanzamiento

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | 01 | **Titulo** | Acceso a la página de inicio de Estudiantes | **Estimación** | 4 horas |
| **Historia de Usuario:** Como estudiante, quiero acceder a una página de inicio informativa que me presente los servicios disponibles en Estudiantes, para poder navegar fácilmente a las secciones que me interesan. | | | | | |
| **Criterios de Aceptación:**  **Escenario 1:**  Dado que el estudiante ingresa a la plataforma “Portal Estudiantil EPN”, cuando inicia la página, entonces se mostrará la página de inicio.  **Escenario 2:**  Dado que el estudiante visualiza la página de inicio, cuando observa el contenido principal, donde se describen los servicios, entonces podrá ver secciones (Cafetería, Polibus, Asosiación, Polimarket) con una breve explicación y un enlace para acceder a cada uno.  **Escenario 3:**  Dado que el estudiante está en la página de inicio, cuando busca opciones de servicios ubicada en el encabezado de la parte derecha, donde se encuentra el menú principal, entonces podrá acceder a él para navegar a las diferentes secciones de la plataforma. | | | | | |
| **Tareas:** | | | | | |
| **T1.** | Diseñar la estructura y el contenido de la página de inicio (Estimación: 1,5 hora). | | | | |
| **T2.** | Desarrollar la página de inicio con HTML y CSS (Estimación: 1.5 horas). | | | | |
| **T3.** | Implementar la funcionalidad de los botones o enlaces para acceder a las diferentes secciones (Estimación: 1 hora). | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | 02 | **Titulo** | Visualización de menús de desayuno y almuerzo | **Estimación** | 6 horas |
| **Historia de Usuario:** Como estudiante, quiero visualizar el menú de desayunos y almuerzos disponibles, para elegir lo que quiero consumir en la cafetería. | | | | | |
| **Criterios de Aceptación:**  **Escenario 1:**  **Dado** que el estudiante accede a la sección de Servicios, **cuando** se dirija a la opción de “Cafetería”, **entonces** se mostrará un menú separado en “Desayunos”, “Almuerzos” y “Bebidas y Snacks”.  **Escenario 2:**  **Dado** que el estudiante visualiza la lista de opciones de almuerzo, **cuando** selecciona un plato específico, **entonces** se desplegarán detalles adicionales del plato.  **Escenario**  **Dado** que el estudiante accede a la sección de Cafetería, **cuando** se despliegue el menú con las categorías de platillos (Desayunos, Almuerzos, Bebidas y Snacks), **entonces** se indicará la disponibilidad de cada categoría en función del horario actual. | | | | | |
| **Tareas:** | | | | | |
| **T1.** | Crear la vista de selección de menú (Estimación: 1 hora). | | | | |
| **T2.** | Implementar la funcionalidad para mostrar las opciones de desayuno y almuerzo (Estimación: 1 hora). | | | | |
| **T3.** | Configurar vista de detalles del plato (Estimación: 1.5 horas). | | | | |
| **T4.** | Mostrar información adicional del plato (ingredientes y tipo) al hacer clic en el plato (Estimación: 1 hora). | | | | |
| **T5.** | Configurar la lógica para determinar si una categoría “Desayuno” o “Almuerzo” está disponible por su horario (Estimación: 1 hora). | | | | |
| **T6.** | Implementar el mensaje de "No hay opciones disponibles" cuando no este disponible (Estimación: 0.5 horas). | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | 03 | **Titulo** | Visualización de precios de los platos de la cafetería | **Estimación** | 5 horas |
| **Historia de Usuario:** Como usuario, quiero ver los precios de cada plato disponible en la cafetería, para poder decidir qué pedir sin sorpresas en el costo. | | | | | |
| **Criterios de Aceptación:**  **Escenario 1:**  **Dado** que el estudiante accede a la sección de la cafetería, **cuando** se despliegue el menú con las categorías de platillos (Desayunos, Almuerzos, Bebidas y Snacks), **entonces** se mostrará el precio del respectivo de cada platillo  **Escenario 2:**  **Dado** que el estudiante accede a la sección de la cafetería , **cuando**  se despliegue el menú con las categorías de platillos (Desayunos, Almuerzos, Bebidas y Snacks) , **entonces** se mostrará el precio actual del plato cuando no esté disponible.  **Escenario 3:**  **Dado** que el estudiante accede a la sección de la cafetería, **cuando**  se despliegue el menú con las categorías de platillos (Desayunos, Almuerzos, Bebidas y Snacks) , **entonces** se mostrará el precio actual del plato cuando esté disponible. | | | | | |
| **Tareas:** | | | | | |
| **T1.** | Crear la vista de selección de menú (Estimación: 1 hora). | | | | |
| **T2.** | Implementar la funcionalidad para mostrar las opciones de desayuno y almuerzo (Estimación: 1 hora). | | | | |
| **T3.** | Configurar vista de detalles del plato (Estimación: 1.5 horas). | | | | |
| **T4.** | Mostrar información adicional del plato (ingredientes y tipo) al hacer clic en el plato (Estimación: 1 hora). | | | | |
| **T5.** | Implementar el mensaje de "No disponible ahora" el plato este fuera de horario (Estimación: 0.5 horas). | | | | |

|  |  |  |
| --- | --- | --- |
| **No.** | **Título** | **Estimación** |
| 01 | Acceso a la página de inicio de PoliEstudiantes | 4h |
| 02 | Visualización de menús de desayuno y almuerzo | 6h |
| 03 | Visualización de precios de los platos de la cafetería | 5h |
|  | **Estimado** | 15h |
| **Real** | 40h |
| **Velocidad** | 0,375 |

**Iteración 2 (Semana 2)**

* **Objetivo Comercial de la Iteración:** Expandir la funcionalidad de la plataforma para incluir información detallada sobre las asociaciones estudiantiles, permitiendo a los estudiantes visualizar los planes y detalles de cada asociación para que puedan decidir si desean unirse a una. Esto facilita la interacción y el conocimiento de los beneficios que cada asociación ofrece, promoviendo una mayor participación en actividades estudiantiles.

**Historias de Usuario (HU)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | 01 | **Titulo** | Visualización de planes de una asociación | **Estimación** | 10 horas |
| **Historia de Usuario:** Como estudiante, quiero visualizar una lista de los planes disponibles para cada asociación, para decidir si alguno de ellos me interesa. | | | | | |
| **Criterios de Aceptación:**  **Escenario 1: Dado** que estoy en la sección de planes de una asociación, **cuando** cargue toda la página, **entonces** se mostrarán los planes disponibles.  **Escenario 2:  Dado** que estoy visualizando la lista de planes, **cuando** ordeno la lista por precio descendente, **entonces** los planes se muestran ordenados de mayor a menor precio.  **Escenario 3: Dado** que estoy visualizando la lista de planes, **cuando** ordeno la lista por precio ascendente, **entonces** los planes se muestran ordenados de menor a mayor precio. | | | | | |
| **Tareas:** | | | | | |
| **T1.** | Desarrollar componente para listar planes (Estimación: 2 horas) | | | | |
| **T2.** | Implementar funcionalidad de selección de plan (Estimación: 3 horas) | | | | |
| **T3.** | Mostrar detalles del plan seleccionado (Estimación: 5 horas) | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | 02 | **Titulo** | Detalles de la asociación | **Estimación** | 8 horas |
| **Historia de Usuario:** Como estudiante, quiero poder ver información detallada de la asociación para entender mejor su misión, servicios y contacto. | | | | | |
| **Criterios de Aceptación:**  **Escenario 1: Dado** que accedo a la sección de asociación, **cuando** cargue la página, **entonces** se mostrará información detallada sobre su misión, servicios y datos de contacto.  **Escenario 2: Dado** que veo los detalles de una asociación, **cuando** hago clic en un enlace a un recurso adicional (redes sociales), **entonces** se abrirá el enlace en una nueva pestaña.  **Escenario 3: Dado** que veo los detalles de una asociación, **cuando** me ubique en el enlace a la página oficial de la FIS y de clic, **entonces** se abrirá el enlace en una nueva pestaña. | | | | | |
| **Tareas:** | | | | | |
| **T1.** | Crear componente para mostrar detalles de la asociación (Estimación: 3 horas) | | | | |
| **T2.** | Crear enlace a las redes sociales de la asociación (Estimación: 2 horas) | | | | |
| **T3.** | Implementar enlace de contacto (Estimación: 3 horas) | | | | |

|  |  |  |
| --- | --- | --- |
| **No.** | **Título** | **Estimación** |
| 01 | Visualización de planes de una asociación | 10h |
| 02 | Detalles de cada asociación | 8h |
|  | **Estimado** | 18h |
| **Real** | 40h |
| **Velocidad** | 0,45 |

**Iteración 3 (Semana 3)**

* **Objetivo Comercial de la Iteración:** Mejorar la experiencia de movilidad de los estudiantes al proporcionar información precisa y actualizada sobre las rutas y horarios del Polibus. Se busca optimizar el tiempo de los estudiantes y facilitar su acceso al transporte universitario, fomentando el uso del Polibus.

**Historias de Usuario (HU)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | 01 | **Titulo** | Visualización de la ruta del Polibus | **Estimación** | 7 horas |
| **Historia de Usuario:** Como estudiante, quiero ver el recorrido completo del Polibus en un mapa dentro de la aplicación, para conocer las rutas exactas que puedo tomar y saber cuál es la parada más cercana a mi ubicación. | | | | | |
| **Criterios de Aceptación:**  **Escenario 1:**  **Dado** que ingreso a la sección de rutas del Polibus en la aplicación, **cuando** selecciono una ruta específica, **entonces** se desplegará un mapa con el recorrido de la ruta que tomará el bus.  **Escenario 2:**  **Dado** que el estudiante se encuentra en la sección del Polibus, **cuando** ingrese en la barra de búsqueda de rutas por nombre o paradas, **entonces** se mostrar la ruta o paradas asociadas a su búsqueda.  **Escenario 3: Dado** que veo la ruta en el mapa, **cuando** selecciono una parada específica en él mapa, **entonces** se mostrará información de la ubicación sobre la parada seleccionada. | | | | | |
| **Tareas:** | | | | | |
| **T1.** | Integrar mapa (Estimación: 1,5 horas) | | | | |
| **T2.** | Mostrar ruta seleccionada en el mapa (Estimación: 2 horas) | | | | |
| **T3.** | Implementar función para seleccionar paradas o rutas (Estimación: 2 horas) | | | | |
| **T4.** | Mostrar información de la parada seleccionada (Estimación: 1,5 horas) | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | 05 | **Titulo** | Visualización del cronograma del Polibus | **Estimación** | 8 horas |
| **Historia de Usuario:** Como estudiante, quiero consultar el cronograma de horarios del Polibus para poder saber a qué hora pasará por mi parada y así organizar mejor mis tiempos. | | | | | |
| **Criterios de Aceptación:**  **Escenario 1:  Dado** que estoy en la sección del Polibus, **cuando** se coloca en carga la tabla de rutas, **entonces** se observa una lista con el nombre de la ruta.  **Escenario 2: Dado** que estoy en la sección de Polibus, **cuando** me dirija a la sección de las rutas, **entonces** se mostrará todos los horarios de cada ruta.  **Escenario 3:  Dado** que estoy consultando la ruta del polibus, **cuando** quiero ver las paradas, **entonces** se observará las paradas por las que pasará el polibus. | | | | | |
| **Tareas:** | | | | | |
| **T1.** | Crear componente para mostrar el cronograma (Estimación: 2 horas) | | | | |
| **T2.** | Cargar datos de horarios desde una fuente externa (Estimación: 2 horas) | | | | |
| **T2.** | Cargar datos de rutas y paradas (Estimación: 2 hora) | | | | |
| **T4.** | Mostrar paradas asociadas a cada ruta (Estimación: 2 horas) | | | | |

|  |  |  |
| --- | --- | --- |
| **No.** | **Título** | **Estimación** |
| 01 | Visualización de la ruta del Polibus | 7h |
| 02 | Visualización del cronograma del Polibus | 8h |
|  | **Estimado** | 15h |
| **Real** | 40h |
| **Velocidad** | 0,375 |

**Iteración 4 (Semana 4)**

* **Objetivo Comercial de la Iteración:** Como estudiante de la Escuela Politécnica Nacional, quiero una plataforma digital que me permita publicar productos en venta, buscar artículos disponibles, filtrar por precio y explorar publicaciones de otros estudiantes, de modo que pueda adquirir fácilmente artículos que me ayuden a mejorar mi experiencia académica y personal, encontrando herramientas, materiales o productos a precios accesibles, además de apoyar la economía estudiantil y fomentar el intercambio entre compañeros.

**Historias de Usuario (HU)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | 01 | **Titulo** | Publicar productos en venta | **Estimación** | 12 horas |
| **Historia de Usuario:** Como estudiante de la Escuela Politécnica Nacional, quiero poder publicar productos en venta mediante un proceso sencillo donde se me pida ingresar el título, descripción, precio, número de contacto, duración de la publicación, además de mi nombre y código de estudiante, para asegurar que las publicaciones provienen de estudiantes verificados y facilitar tanto el ingreso de mis artículos en la plataforma como su adquisición por parte de otros compañeros de la institución. | | | | | |
| **Criterios de Aceptación (INVEST):**  **Escenario 1:**  **Dado** que el estudiante está en la sección Polimarket, **cuando** selecciona “Vender”, **entonces** se mostrara un formulario para ingresar los datos del producto.  **Escenario 2:**  **Dado** que el estudiante ha ingresado todos los datos del producto, **cuando** selecciona “Publicar”, **entonces** el sistema guarda el producto.  **Escenario 3:**  **Dado** que el estudiante intenta publicar sin ingresar datos obligatorios, **cuando** selecciona “Publicar”, **entonces** el sistema muestra mensajes de error” Por favor, completa todos los campos.”. | | | | | |
| **Tareas:** | | | | | |
| **T1.** | Diseñar el formulario de publicación (Estimación: 3 horas) | | | | |
| **T2.** | Implementar la lógica del controlador para procesar la publicación (Estimación: 5 horas) | | | | |
| **T3.** | Crear la clase Producto con atributos y métodos necesarios (Estimación: 4 horas) | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | 02 | **Titulo** | Visualizar Publicaciones de Productos en Venta | **Estimación** | 7 horas |
| **Historia de Usuario:** Como estudiante de la Escuela Politécnica Nacional, quiero poder visualizar las publicaciones de productos en venta de otros estudiantes, mostrando el título, precio, nombre del vendedor, número de contacto y el tiempo restante de la publicación, para poder identificar rápidamente los artículos que me interesen y contactar al vendedor de manera fácil y directa. | | | | | |
| **Criterios de Aceptación (INVEST):**  **Escenario 1:**  **Dado** que el estudiante está en la plataforma, **cuando** la página se carga, **entonces** se muestra una lista de productos publicados recientemente.  **Escenario 2:**  **Dado** que el estudiante está visualizando la lista de productos en venta, **cuando** se agregue un producto, **entonces** la página mostrara el nuevo producto agregado.  **Escenario 3:**  **Dado** que el estudiante está visualizando un producto en la lista de ventas, **cuando** hace clic en el número de contacto, **entonces** el sistema muestra un enlace o acción para contactar al vendedor mediante WhatsApp. | | | | | |
| **Tareas:** | | | | | |
| **T1.** | Crear la vista para mostrar la lista de productos en venta (Estimación: 3 horas) | | | | |
| **T2.** | Crear el mensaje para indicar que no existen productos (Estimación: 2 horas) | | | | |
| **T3.** | Implementar la funcionalidad para que el estudiante pueda contactar al vendedor mediante WhatsApp al hacer clic en el número de contacto (Estimación: 2 horas) | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | 03 | **Titulo** | Buscar y filtrar Producto en Venta | **Estimación** | 12 horas |
| **Historia de Usuario:** Como estudiante de la Escuela Politécnica Nacional, quiero poder buscar y filtrar los productos en venta por precio, para poder encontrar rápidamente los artículos que me interesen y ajusten a mi presupuesto. | | | | | |
| **Criterios de Aceptación (INVEST):**  **Escenario 1:**  **Dado** que el estudiante está visualizando los productos disponibles en venta, **cuando** aplique el filtro de búsqueda e ingresa el nombre de un producto, **entonces** el sistema mostrará todos los productos cuyo título coincida con el texto ingresado, sin tener en cuenta el precio.  **Escenario 2:**  **Dado** que el estudiante está visualizando los productos en venta, **cuando** el estudiante aplica un filtro de búsqueda por precio, **entonces** el sistema muestra solo los productos cuyo precio se ajuste al rango especificado, sin mostrar los que queden fuera del rango.  **Escenario 3:**  **Dado** que el estudiante está visualizando los productos disponibles en venta, **cuando** aplica el filtro de búsqueda y realiza una búsqueda ingresando un nombre de producto, y luego ajusta el filtro de precio utilizando el rango deslizante, **entoncces** el sistema mostrará únicamente los productos cuyo título coincida con la búsqueda y cuyo precio esté dentro del rango seleccionado. | | | | | |
| **Tareas:** | | | | | |
| **T1.** | Diseñar e implementar el campo de búsqueda por título en la interfaz de usuario, permitiendo al estudiante ingresar texto (Estimación: 3 horas). | | | | |
| **T2.** | Diseñar e implementar un filtro de precio con un control de rango deslizante, para que los estudiantes puedan especificar un rango de precio de manera intuitiva (Estimación: 4 horas). | | | | |
| **T3.** | Implementar la lógica de búsqueda y filtrado en el backend para que el sistema devuelva resultados basados en el título y/o rango de precio seleccionados (Estimación: 5 horas). | | | | |

|  |  |  |
| --- | --- | --- |
| **No.** | **Título** | **Estimación** |
| 1 | Publicar productos en venta | 12h |
| 2 | Visualizar Publicaciones de Productos en Venta | 7h |
| 3 | Buscar y filtrar Producto en Venta | 12h |
|  | **Estimado** | 31 h |
| **Real** | 40h |
| **Velocidad** | 0,775 |

# 2. Flujo de Calidad del Código

Para asegurar la calidad del software en el proyecto **PoliCampus**, se definen scripts automatizados y prácticas de desarrollo que se pueden ejecutar manualmente desde consola o integrarse a pipelines de integración continua (CI) como **Jenkins**.

**2.1 Compilación**

La compilación del proyecto se realiza usando **Apache Maven**, el sistema de construcción estándar para proyectos Java.

**Script de compilación:**

mvn clean install

Este comando:

* Limpia el proyecto (clean)
* Descarga dependencias (dependency:resolve)
* Compila el código (compile)
* Ejecuta pruebas (test)
* Empaqueta el proyecto (package)

El archivo de configuración pom.xml incluye todos los módulos del sistema: cafetería, polibus, asociaciones y PoliMarket.

**2.2 Linter (Verificación de estilo y buenas prácticas)**

Se usa **Checkstyle** para validar convenciones de código Java.

Configuración en Maven (pom.xml):

<plugin>

  <groupId>org.apache.maven.plugins</groupId>

  <artifactId>maven-checkstyle-plugin</artifactId>

  <version>3.1.2</version>

  <configuration>

    <configLocation>google\_checks.xml</configLocation>

    <failOnViolation>true</failOnViolation>

  </configuration>

  <executions>

    <execution>

      <phase>verify</phase>

      <goals><goal>check</goal></goals>

    </execution>

  </executions>

</plugin>

**Ejecución desde consola:**

mvn checkstyle:check

**2.3 Pruebas Unitarias y/o de Integración**

El sistema utiliza **JUnit 5** para pruebas unitarias e integradas. También se emplea **Mockito** para mocks y pruebas con dependencias externas simuladas.

**Ejecutar pruebas:**

mvn test

Se integran estas pruebas en el pipeline Jenkins para validar cada Pull Request.

**Cobertura recomendada:**

* Controladores: pruebas con entradas válidas e inválidas
* Servicios: validación de lógica de negocio
* DAO o repositorios: pruebas con H2 en memoria

Estructura de paquetes para pruebas:

src/

 └── test/

      ├── java/

      │    └── ec.epn.policampus.\*

      └── resources/

**2.4 Dockerfile para Despliegue**

El despliegue se realiza mediante contenedores Docker. La aplicación web (WAR generado por Maven) se ejecuta dentro de un servidor **Apache Tomcat 9**.

**Dockerfile para PoliCampus:**

FROM tomcat:9-jdk17

LABEL maintainer="equipo@policampus.epn.edu.ec"

# Directorio del servidor

WORKDIR /usr/local/tomcat/webapps/

# Copiar el WAR generado por Maven

COPY target/policampus.war ./ROOT.war

# Exponer el puerto de la aplicación

EXPOSE 8080

# Comando para ejecutar Tomcat

CMD ["catalina.sh", "run"]

**Comandos de uso:**

docker build -t policampus-app .

docker run -d -p 8080:8080 --name policampus policampus-app

# 3. Conclusiones

El desarrollo de la plataforma **PoliCampus** permitió implementar una solución web funcional y modular, pensada para mejorar la experiencia universitaria de los estudiantes de la Escuela Politécnica Nacional. A través de una arquitectura bien organizada y el uso de herramientas estándar como Java, Maven, Jenkins y Docker, se logró cumplir con los objetivos técnicos y funcionales del proyecto.

Las funcionalidades implementadas (módulo de bienvenida, cafetería, Polibus, asociaciones y PoliMarket) reflejan un enfoque centrado en las necesidades reales de los usuarios finales. El equipo aplicó buenas prácticas de desarrollo ágil, integración continua y pruebas automatizadas, lo cual garantiza escalabilidad, mantenibilidad y facilidad de despliegue.

# 4. Recomendaciones

* Fortalecer la cultura DevOps en el equipo, manteniendo pipelines CI/CD activos para todas las futuras versiones, incluyendo pruebas automáticas y análisis de código estático.
* Mantener la documentación técnica actualizada, especialmente el README.md, los diagramas de arquitectura, el Dockerfile y cualquier script de despliegue.
* Implementar monitoreo en producción (logs, métricas de uso y errores) para mejorar el soporte post-despliegue y la toma de decisiones basada en datos reales.
* Aumentar progresivamente la cobertura de pruebas unitarias e integradas, apuntando a al menos un 80 % del código cubierto y considerando pruebas de UI automatizadas si se incorpora frontend complejo.
* Planificar ciclos regulares de mantenimiento, con énfasis en revisiones de seguridad, actualización de dependencias y mejora de la experiencia de usuario.
* Incluir retroalimentación directa de los usuarios reales (estudiantes de la EPN) para futuras versiones, incorporando sus sugerencias en el backlog del producto.