**Departamento de Ciencias de la Computación (DCCO)**

**Carrera de Ingeniería de Software**

**Prácticas Pre Profesionales (PPP)**

Perfil del Proyecto

Presentado por: Limones Gavilanes John Jairo

Tutor académico: Ruiz Robalino Jenny

Ciudad: Quito, Ecuador

Fecha: 29/07/2025

**Índice Pág.**

PERFIL DE PROYECTO

1. Introducción….

2. Planteamiento del trabajo….

2.1 Formulación del problema….

2.2 Justificación….

3. Sistema de Objetivos….

3.1. Objetivo General……..

3.2. Objetivos Específicos (03)

4. Alcance….

5.1 Metodología (Marco de trabajo 5W+2H) ….

6. Ideas a Defender ….

7. Resultados Esperados

8. Viabilidad(Ej.) …..

8.1 Humana….

8.1.1 Tutor Empresarial ….

8.1.2 Tutor Académico….

8.1.3 Estudiantes….

8.2 Tecnológica….

8.2.1 Hardware….

8.2.2 Software….

9. Cronograma: ….

10. Bibliografía….

1. **Introducción**

En las diferentes sedes de la Universidad de las Fuerzas Armadas ESPE se realizan constantemente actividades académicas, investigativas y administrativas que requieren la reserva de espacios físicos y el uso de artículos o equipos específicos. Actualmente, este proceso se lleva a cabo de forma manual o mediante canales no centralizados, lo que genera desorganización, retrasos y dificultad para llevar un control adecuado. Esta situación evidencia una clara necesidad de modernizar el sistema de gestión, creando una herramienta tecnológica que permita unificar y optimizar estos procedimientos.

El proyecto ESPE-REG surge como una respuesta a esta problemática, con el propósito de desarrollar un sistema web que facilite la administración de reservas en las tres sedes principales (Matriz, Latacunga y Santo Domingo). Su implementación busca ofrecer a los usuarios una plataforma intuitiva, segura y centralizada que permita mejorar la eficiencia de los procesos internos, reducir tiempos de gestión y asegurar una trazabilidad adecuada de todas las solicitudes y recursos disponibles.

1. **Planteamiento del trabajo**

**2.1 Formulación del problema**

El proyecto ESPE-REG plantea desarrollar un sistema web centralizado que automatice la gestión de reservas de espacios y artículos en las tres sedes de la ESPE. La propuesta busca sustituir los procesos manuales actuales, que generan desorden y retrasos, por una plataforma moderna que permita a los usuarios realizar solicitudes en línea y a los encargados y administradores controlar y auditar cada acción de manera eficiente.

**2.2 Justificación**

La implementación de ESPE-REG es relevante porque introduce procesos automatizados y centralizados en la gestión de recursos institucionales, unificando procedimientos que hoy son manuales y dispersos. Este proyecto aporta un modelo escalable y replicable que puede servir de base para futuras investigaciones en sistemas de información académicos, optimización de procesos y auditoría digital, generando un impacto positivo tanto en la gestión interna como en el ámbito académico y tecnológico..

1. **Sistema de Objetivos**

**3.1. Objetivo General**

El proyecto ESPE-REG tiene como objetivo desarrollar un sistema web que centralice y automatice la gestión de reservas de espacios y artículos en las sedes Matriz, Latacunga y Santo Domingo de la ESPE. Para lograrlo, se emplearán tecnologías modernas como Laravel 12, MySQL, Breeze y Spatie, a fin de ofrecer una plataforma segura, eficiente y trazable que optimice los procesos actuales y mejore el control institucional.

**3.2. Objetivos Específicos (03)**

* Implementar un sistema de autenticación y roles que permita controlar los accesos y privilegios según el perfil de usuario (administrador, encargado y usuario).
* Desarrollar módulos para la gestión de reservas y recursos (lugares y artículos) de manera centralizada y en tiempo real.
* Incorporar un sistema de auditoría que registre todas las acciones realizadas en la plataforma, garantizando la transparencia y trazabilidad del proceso.

1. **Alcance**

El sistema permitirá:

* La autenticación de usuarios únicamente con correos institucionales.
* La gestión de lugares, artículos y solicitudes de reservas.
* La aprobación o rechazo de solicitudes por parte de administradores y encargados.
* La generación de registros de auditoría automáticos.
* La visualización personalizada de información y funciones de acuerdo con el rol y la sede seleccionada.

1. **Marco Teórico**

El proyecto ESPE-REG se fundamenta en el uso de tecnologías modernas que permiten construir aplicaciones web robustas, seguras y escalables. Para el desarrollo del backend se empleará Laravel 12, un framework de PHP ampliamente utilizado por su arquitectura organizada y su capacidad de integrar de forma nativa buenas prácticas de programación. La gestión de datos estará respaldada por MySQL, una base de datos relacional que facilita el almacenamiento estructurado y confiable de la información.

Para el control de usuarios y permisos se utilizará la librería Spatie, que ofrece una administración detallada de roles, lo que resulta fundamental para diferenciar las funcionalidades según el tipo de usuario (administrador, encargado o usuario). Además, la plantilla Breeze permitirá implementar de manera rápida y segura el módulo de autenticación, mientras que Tailwind CSS se usará para diseñar interfaces responsivas y modernas, favoreciendo la experiencia del usuario.

El entorno de desarrollo se llevará a cabo sobre XAMPP, que ofrece un servidor local con soporte para PHP y MySQL, lo que facilita la fase de construcción y pruebas antes de implementar el sistema en producción.

**5.1 Metodología (Marco de trabajo 5W+2H)**

Debe explicar paso a paso el desarrollo de la guía con la herramienta de Excel aplicando el marco de trabajo de las 5W y 2H

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ¿QUÉ? | ¿CÓMO? | ¿QUIÉN? | ¿CUÁNDO? | ¿POR QUÉ? | ¿CUÁNTO? | % DE CUMPLIMIENTO |
| Sistema web de gestión de reservas multi-sede | Usando Laravel 12, MySQL, Spatie y Breeze bajo metodología incremental | Estudiante de PPP con tutor académico | Julio – Agosto 2025 | Optimizar la gestión y control de recursos | Con costo – 2 meses | 80% (proceso) |

Tabla 1 Marco de trabajo 5W+2H

1. **Ideas a Defender**

El desarrollo del sistema ESPE-REG se basa en la idea de que la tecnología puede transformar procesos administrativos complejos en procedimientos más eficientes, seguros y transparentes. La propuesta defiende el uso de buenas prácticas de la ingeniería de software aplicadas desde el análisis y diseño hasta la implementación, aprovechando principios de programación estructurada y orientada a objetos.

Otra idea clave es la incorporación de un control estricto de roles y permisos que permita diferenciar las responsabilidades de cada usuario dentro del sistema, asegurando la integridad y trazabilidad de la información. Finalmente, se sostiene la importancia de ofrecer una plataforma centralizada, accesible y escalable, capaz de evolucionar y adaptarse a nuevas necesidades institucionales en el futuro.

1. **Resultados Esperados**

Como resultado del proyecto se espera entregar un sistema web funcional que automatice y centralice la gestión de reservas de espacios y artículos en las sedes de la ESPE. Con esta herramienta se busca reducir los tiempos de gestión, evitar errores derivados de los métodos manuales y mejorar la trazabilidad de las acciones realizadas. Además, se espera que el proyecto permita a los estudiantes aplicar de forma práctica los conocimientos adquiridos en fundamentos de ingeniería de software y programación, generando un producto tecnológico de impacto académico y administrativo.

|  |  |  |  |
| --- | --- | --- | --- |
| Cantidad | Descripción | Valor Unitario (USD) | Valor Total (USD) |
|  | **Equipo en casa** |  |  |
| 1 | Laptop Hp pro-book/ 16gb RAM / 256gb SSD | 600 | 600 |
|  |  |  |  |
|  | **Software** |  |  |
| 1 | Sistema operativo Windows 10 | 145 | 145 |
| 1 | Visual Studio Code | 0 | 0 |
| 1 | XAMP | 0 | 0 |
| 1 | FileZilla | 0 | 0 |
|  | | TOTAL | 745 |

1. **Viabilidad(Ej.)**

Tabla 2 Presupuesto del proyecto

Debe explicar los recursos necesarios para su proyecto y adicionalmente la viabilidad del punto 8.1. y 8.2

**8.1 Humana**

**8.1.1 Tutor Empresarial**

**Ing. Carlos**

**• Analista UTICS**

**8.1.2 Tutor Académico**

**Ing. Jenny Ruiz**

* **Avance del proceso de practicas PP**

**8.1.3 Estudiante**

**Limones John**

**• Desarrollo de la aplicación**

**8.2 Tecnológica**

**8.2.1 Hardware**

|  |  |  |
| --- | --- | --- |
|  | Requisitos mínimos | Disponibilidad |
| Memoria RAM | 8 GB de RAM | Alta |
| Almacenamiento | 10 GB de espacio de almacenamiento | Alta |

Tabla 3 Requisitos de Hardware

**8.2.2 Software**

|  |  |  |
| --- | --- | --- |
|  | Requisitos mínimos | Disponibilidad |
| Sistema Operativo | Se recomienda Windows 10 u 11, macOS  10.10 o Ubuntu 16 | Alta |
| IDE | Es recomendable Visual Studio Code debido a su conexión con FTP, sin embargo, cualquier IDE con esta funcionalidad funciona. | Alta |

Tabla 4 Requisitos de Software

1. **Conclusiones y recomendaciones**

## **9.1 Conclusiones**

El desarrollo del sistema ESPE‑REG permite evidenciar cómo la aplicación de principios de ingeniería de software y programación puede transformar procesos administrativos tradicionales en soluciones modernas y eficientes. A lo largo del proyecto se han cumplido los objetivos planteados: se ha diseñado una plataforma web que centraliza la gestión de reservas en las tres sedes de la ESPE, incorporando control de roles, seguridad en el acceso y registro de auditorías para garantizar la trazabilidad de todas las acciones.

Esta solución representa un aporte significativo, ya que optimiza tiempos, reduce errores derivados de los procesos manuales y genera un modelo tecnológico escalable que puede ser replicado en otros ámbitos institucionales.

## **9.2 Recomendaciones**

Se recomienda mantener la actualización constante de la plataforma y capacitar a los usuarios en su uso para aprovechar al máximo sus funcionalidades. También se sugiere, en futuras versiones, incorporar reportes estadísticos, notificaciones automáticas y módulos adicionales que permitan ampliar la cobertura del sistema. Finalmente, se aconseja evaluar periódicamente el rendimiento y la seguridad del sistema para asegurar su continuidad y mejora a largo plazo.

.

1. **Planificación para el Cronograma:**

Debe insertar una imagen clara y legible de la planificación del proyecto a desarrollar.

|  |  |  |  |
| --- | --- | --- | --- |
| FECHA | ACTIVIDAD | TAREA | HORAS DE TRABAJO |
| 2025-06-20 | Capacitación y orientación de funciones, familiarización con tecnologías ( Spring Boot y Flutter ) | Se investigan y analizan las tecnologías clave, como Spring Boot para el backend y Flutter para el frontend, para comprender sus capacidades, fortalezas y cómo se integran con otras herramientas. | 2 |
| Se define el proceso de desarrollo, incluyendo la metodología previa a la investigación del proyecto | 2 |
| Se estudian y adoptan estándares de codificación | 1 |
| Se instalan y configuran las herramientas necesarias, como JDK, IDEs (como IntelliJ o VS Code), y SDKs (Flutter), para crear un entorno de trabajo funcional y eficiente. | 1 |
| 2025-06-23 | Análisis de Spring Boot, Flutter con manejo de bases de datos en ejemplos de API | Revisión de funcionalidades clave en Spring Boot y Flutter. | 2 |
| Configuración de entornos y compatibilidad de componentes. | 2 |
| Práctica con ejemplos básicos de bases de datos y APIs. | 2 |
| 2025-06-24 | Conexión y gestión de base de datos MySQL en XAMPP. | 2 |
| Diseño de APIs simples en Spring Boot para pruebas iniciales. | 2 |
| Aplicación de principios SOLID en el diseño del código. | 2 |
| 2025-06-25 | Proyecto CRUD Usuarios (Spring Boot/IntelliJ) con xampp | Diseño del esquema de la base de datos para usuarios (tablas, campos, etc.). | 3 |
| Configuración de la base de datos MySQL en XAMPP. | 2 |
| Creación del proyecto de Spring Boot y configuración de dependencias. | 1 |
| 2025-06-26 | Desarrollo de la API REST para usuarios en Spring Boot (CRUD básico). | 6 |
| 2025-06-27 | Desarrollo de la interfaz de usuario en Flutter para el CRUD de usuarios. | 6 |
| 2025-06-30 | Integración completa y refinamiento del proyecto CRUD de usuarios | 2 |
| Sincronización de Spring Boot y Flutter | 2 |
| Aplicación de conceptos de Docker vistos previamente. Validación final de funcionalidad | 2 |
| 2025-07-01 | Proyecto Microservicios "Encuentro" con Eureka Server | Inicio del proyecto Encuentro: configuración inicial de microservicios y preparación del entorno. | 3 |
| Implementación de Eureka Server para el descubrimiento de servicios. | 3 |
| 2025-07-02 | Desarrollo del primer microservicio (usuarios) y registro en Eureka Server. | 3 |
| Desarrollo del segundo microservicio (productos/servicios) y conexión con el de usuarios. | 3 |
| 2025-07-03 | Configuración de la comunicación entre microservicios mediante Eureka. | 3 |
| Implementación de pruebas básicas de interacción entre los servicios desarrollados. | 3 |
| 2025-07-04 | Integración de un sistema de monitoreo básico para supervisar el estado de los microservicios. | 3 |
| Diseño de un API Gateway para centralizar el acceso y mejorar la seguridad del proyecto Encuentro. | 3 |
| 2025-07-07 | Estudio e Investigación: Módulo Reconocimiento QR | Investigación inicial de bibliotecas y APIs para QR en Spring Boot, sin implementación completa. | 3 |
| Revisión de alternativas para reconocimiento QR en proyectos Java, aún en análisis. | 3 |
| 2025-07-08 | Desarrollo parcial de un prototipo básico de reconocimiento QR en Spring Boot. | 3 |
| Pruebas preliminares de funcionamiento del prototipo con end points, sin consolidación. | 3 |
| 2025-07-09 | Análisis de integración de módulos QR en aplicaciones Flutter, en etapa inicial. | 3 |
| Exploración de paquetes y dependencias de Flutter para lectura de QR, sin aplicación práctica. | 3 |
| 2025-07-10 | Estudio de la comunicación entre Flutter y Spring Boot para gestión de QR, con avance limitado. | 6 |
| 2025-07-11 | Investigación de integración de QR en Flutter con end points, sin pruebas concluyentes. | 6 |
| 2025-07-14 | Preparación para el Cambio de Proyecto y análisis para registros | Definir la estructura de la base de datos para roles y permisos. | 3 |
| Configurar el entorno de Laravel para usar Spatie. | 3 |
| 2025-07-15 | Implementar la asignación de roles a usuarios. | 3 |
| Crear la interfaz de usuario para la gestión de roles. | 3 |
| 2025-07-16 | Desarrollar Request Validators para formularios de registro y reserva. | 3 |
| Integrar los validadores en los formularios correspondientes. | 3 |
| 2025-07-17 | Realizar pruebas unitarias de la asignación de roles. | 3 |
| Realizar pruebas de integración de los validadores. | 3 |
| 2025-07-18 | Ajustar la configuración de permisos de Spatie. | 3 |
| Generar pantallas iniciales por rol. | 3 |
| 2025-07-21 | Diseño y Desarrollo de Pantallas Iniciales | Diseñar wireframes y maquetas de las pantallas principales. | 3 |
| Desarrollar la interfaz de usuario para el administrador. | 3 |
| 2025-07-22 | Desarrollar la interfaz de usuario para el usuario regular. | 3 |
| Desplegar la plataforma en un entorno de desarrollo. | 3 |
| 2025-07-23 | Analizar el código de MRBS para la adaptación. | 3 |
| Modificar el código de MRBS para la lógica de negocio de Espe Reg. | 3 |
| 2025-07-24 | Desarrollar la funcionalidad para gestionar el calendario. | 3 |
| Implementar el módulo de gestión de ubicaciones. | 3 |
| 2025-07-25 | Optimizar el diseño de la interfaz de usuario de MRBS. | 3 |
| Adaptar la interfaz de MRBS a la identidad visual de Espe Reg. | 3 |
| 2025-07-28 | Refinamiento de Módulos y Experiencia de Usuario | Realizar ajustes menores en el despliegue de la plataforma. | 3 |
| Optimizar la configuración del servidor para la producción. | 3 |
| 2025-07-29 | Refinar las funcionalidades de registro y reserva. | 3 |
| Revisar la lógica de negocio del proyecto. | 3 |
| 2025-07-30 | Optimizar la eficiencia de las consultas a la base de datos. | 3 |
| Mejorar la interactividad y fluidez de la interfaz de usuario. | 3 |
| 2025-07-31 | Implementar animaciones y transiciones sutiles. | 3 |
| Implementar el diseño de un sistema de notificaciones para usuarios. | 3 |
| 2025-08-01 | Desarrollar componentes reutilizables para la interfaz de usuario (frontend). | 3 |
| Mejorar la experiencia general del usuario. | 3 |
| 2025-08-04 | Preparación del Despliegue y Presentación | Explorar e implementar integraciones con sistemas de calendarios externos. | 3 |
| Optimizar el rendimiento de las consultas de la base de datos. | 3 |
| 2025-08-05 | Revisar las configuraciones de seguridad de la aplicación. | 3 |
| Preparar la base de datos final y los scripts de despliegue. | 3 |
| 2025-08-06 | Realizar las migraciones y Seed de la base de datos. | 3 |
| Cambiar el entorno del proyecto de test a despliegue. | 3 |
| 2025-08-07 | Refinar la interfaz de usuario y la experiencia general. | 3 |
| Preparar la presentación del proyecto como vistas del usuario y administrador. | 3 |
| 2025-08-08 | Incluir demostraciones en vivo en la presentación. | 3 |
| Realizar una primera recolección de posibles modificaciones. | 3 |
| 2025-08-12 | Modificaciones Post-Presentación y Validación(Post-Presentación) | Recopilar y analizar el feedback de la presentación. | 3 |
| Identificar las áreas prioritarias para mejoras y correcciones. | 3 |
| 2025-08-13 | Priorizar la implementación de las primeras modificaciones. | 3 |
| Implementar las correcciones solicitadas. | 3 |
| 2025-08-14 | Implementar las primeras mejoras de interfaz | 3 |
| Realizar pruebas unitarias de los módulos modificados. | 3 |
| 2025-08-15 | Realizar pruebas de integración para asegurar que no haya regresiones. | 2 |
| Validar las modificaciones con los stakeholders. | 2 |
| Generar un informe de las modificaciones implementadas. Reade.me | 2 |

Tabla 5 Cronograma del proyecto.

1. **Referencias**

**Anexos.**

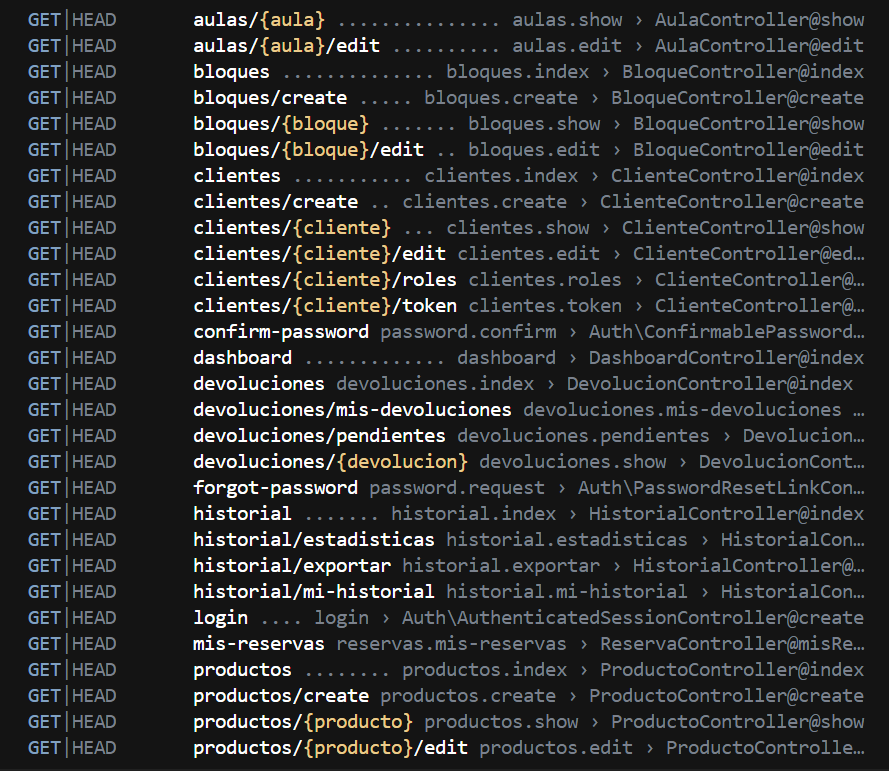
**Anexo 1 – Figura 2**

**Listado de rutas GET de la aplicación de reservas**

A continuación, se presenta el listado de las rutas GET configuradas en la aplicación de reservas, las cuales permiten acceder a las diferentes funcionalidades del sistema desde el frontend o mediante pruebas en el entorno de desarrollo. Este listado facilita la comprensión de la estructura de navegación y los endpoints disponibles para la consulta de información dentro de la plataforma:

***Tabla 2: Lista de rutas GET***

|  |  |  |  |
| --- | --- | --- | --- |
| **Ruta** | **Descripción** | **Método** | **Observaciones** |
| /reservas | Obtiene todas las reservas registradas | GET | Endpoint principal de consulta |
| /reservas/{id} | Obtiene los detalles de una reserva específica | GET | Requiere ID de la reserva |
| /usuarios | Lista todos los usuarios registrados | GET | Endpoint protegido por permisos |
| /usuarios/{id} | Obtiene información de un usuario específico | GET | Requiere ID de usuario |
| /aulas | Lista todas las aulas disponibles para reserva | GET | Permite filtrado por ubicación |
| /calendario | Muestra el calendario de reservas por día/mes | GET | Integrado con la interfaz del calendario |



*Figura 2. Listado de rutas GET de la aplicación de reservas, utilizado para pruebas, consulta de datos y desarrollo de interfaces.*

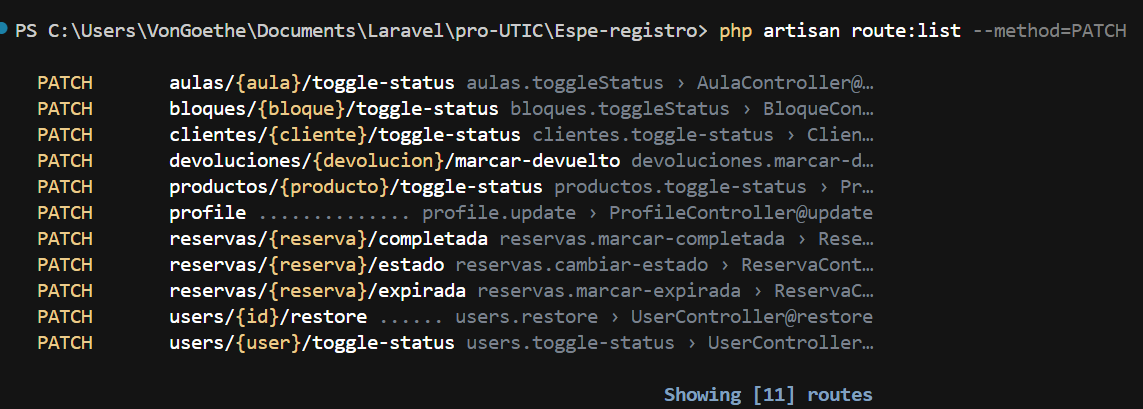
**Anexo 2 – Figura 3**

**Listado de rutas PATCH de la aplicación de reservas**

En la siguiente tabla se presentan las rutas PATCH configuradas en la aplicación de reservas. Estas rutas permiten realizar actualizaciones parciales sobre recursos específicos, tales como cambiar estados, marcar reservas completadas o restaurar usuarios. La documentación de estas rutas contribuye a comprender la estructura de la API y las funcionalidades disponibles para la gestión dinámica de la información.

***Tabla 3: PATCH de la aplicación de reservas***

|  |  |  |  |
| --- | --- | --- | --- |
| **Ruta** | **Nombre del endpoint** | **Controlador / Acción** | **Descripción** |
| /aulas/{aula}/toggle-status | aulas.toggleStatus | AulaController@toggleStatus | Cambia el estado activo/inactivo de un aula |
| /bloques/{bloque}/toggle-status | bloques.toggleStatus | BloqueController@toggleStatus | Cambia el estado activo/inactivo de un bloque |
| /clientes/{cliente}/toggle-status | clientes.toggle-status | ClienteController@toggleStatus | Activa o desactiva un cliente en el sistema |
| /devoluciones/{devolucion}/marcar-devuelto | devoluciones.marcar-devuelto | DevolucionesController@marcarDevuelto | Marca un artículo como devuelto |
| /productos/{producto}/toggle-status | productos.toggle-status | ProductoController@toggleStatus | Cambia el estado activo/inactivo de un producto |
| /profile | profile.update | ProfileController@update | Actualiza los datos del perfil del usuario |
| /reservas/{reserva}/completada | reservas.marcar-completada | ReservaController@marcarCompletada | Marca una reserva como completada |
| /reservas/{reserva}/estado | reservas.cambiar-estado | ReservaController@cambiarEstado | Cambia el estado general de la reserva |
| /reservas/{reserva}/expirada | reservas.marcar-expirada | ReservaController@marcarExpirada | Marca una reserva como expirada |
| /users/{id}/restore | users.restore | UserController@restore | Restaura un usuario previamente eliminado |
| /users/{user}/toggle-status | users.toggle-status | UserController@toggleStatus | Activa o desactiva un usuario en el sistema |



*Figura 3.* *Listado de rutas PATCH de la aplicación de reservas, utilizado para la actualización de estados y gestión dinámica de recursos.*

**Anexo 3 – Figura 4**

**Procedimiento para la creación y gestión de productos en el inventario**

A continuación se detalla el procedimiento utilizado para crear y gestionar productos dentro del sistema de reservas, incluyendo rutas, acciones y buenas prácticas. Esta documentación se adjunta como referencia visual y técnica para el correcto uso del módulo de inventario:

***Tabla 4: Gestión de productos (inventario)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Acción** | **Ruta / Ubicación** | **Descripción** | **Observaciones / Buenas prácticas** |
| Crear producto | productos/create | Permite registrar un nuevo producto en el inventario ingresando su nombre, descripción, stock disponible y estado (activo/inactivo). | Mantener descripciones claras y precisas; indicar unidades de medida; usar “inactivo” para productos sin stock o fuera de uso. |
| Listar productos | productos | Muestra todos los productos registrados en el sistema. | Permite verificar el estado y stock de cada producto. |
| Ver/Editar producto | productos/{producto}, productos/{producto}/edit | Permite consultar la información de un producto específico o modificar sus datos. | Revisar consistencia de los datos antes de guardar cambios. |
| Activar/Desactivar producto | PATCH productos/{producto}/toggle-status | Cambia el estado de un producto entre activo e inactivo según disponibilidad o uso. | Útil para mantener actualizado el inventario sin eliminar registros. |

****

*Figura 4. Interfaz y rutas utilizadas para la creación y gestión de productos en el inventario, incluyendo acciones de activación/desactivación y edición de información.*

**Anexo 4 – Figura 5**

**Procedimiento para la creación y gestión de bloques (franjas horarias)**

El siguiente anexo documenta el procedimiento para crear y administrar bloques de tiempo dentro del sistema de reservas, incluyendo rutas, acciones rápidas y recomendaciones de buenas prácticas para la gestión de horarios:

***Tabla 5: Bloques***

|  |  |  |  |
| --- | --- | --- | --- |
| **Acción** | **Ruta / Ubicación** | **Descripción** | **Observaciones / Buenas prácticas** |
| Crear bloque | bloques/create | Permite registrar un nuevo bloque de tiempo especificando nombre o código, hora de inicio, hora de fin y estado (activo/inactivo). | Validar que los bloques no se solapen con otros existentes. Mantener consistencia en la nomenclatura. |
| Listar/Editar bloque | bloques, bloques/{bloque}/edit | Permite visualizar todos los bloques registrados y modificar los datos de bloques específicos. | Revisar horarios antes de guardar cambios; asegurar que los bloques sean coherentes con la planificación académica. |
| Activar/Desactivar bloque | PATCH bloques/{bloque}/toggle-status | Cambia el estado del bloque entre activo e inactivo según su uso actual. | Mantener activos solo los bloques realmente en uso; estandarizar horarios (p. ej., 07:00–09:00, 09:00–11:00) para facilitar la planificación. |

**

*Figura 6. Interfaz y rutas utilizadas para la creación y gestión de aulas, incluyendo acciones de activación/desactivación y consultas por bloque.*

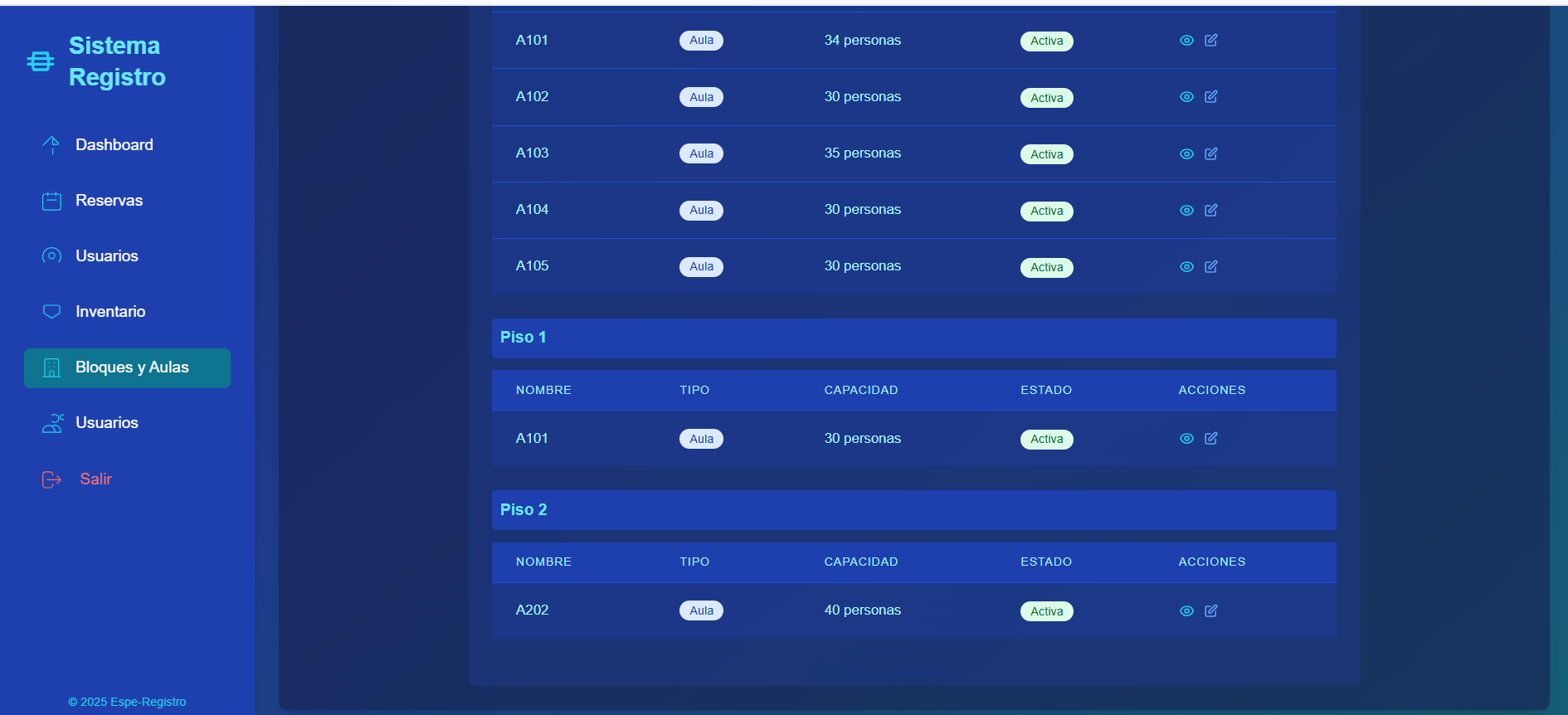
**Anexo 5 – Figura 6**

**Procedimiento para la creación y gestión de aulas**

El siguiente anexo documenta el procedimiento para agregar y administrar aulas dentro del sistema de reservas, incluyendo rutas, acciones rápidas, consultas y recomendaciones de buenas prácticas:

***Tabla 6: Agregar aulas a los bloques***

|  |  |  |  |
| --- | --- | --- | --- |
| **Acción** | **Ruta / Ubicación** | **Descripción** | **Observaciones / Buenas prácticas** |
| Agregar aula | aulas/create | Permite registrar un nuevo aula ingresando su nombre, capacidad, ubicación y estado (activo/inactivo). | Confirmar que el aula se haya guardado correctamente en el listado. Usar nombres únicos y cortos (p. ej., A-101, Lab-Redes). |
| Listar/Editar aula | aulas, aulas/{aula}/edit | Permite visualizar todas las aulas registradas y modificar datos específicos de cada aula. | Mantener la capacidad actualizada para evitar sobrecupos y garantizar una correcta asignación de espacios. |
| Activar/Desactivar aula | PATCH aulas/{aula}/toggle-status | Cambia el estado de un aula entre activo e inactivo según su disponibilidad. | Mantener inactivas las aulas que no estén en uso para reflejar la disponibilidad real. |
| Consultas por bloque | api/aulas/by-bloque | Permite obtener aulas disponibles filtradas por bloques de tiempo específicos mediante la API. | Útil para integraciones con sistemas de reservas y planificación automática. |



*Figura 6. Interfaz y rutas utilizadas para la creación y gestión de aulas, incluyendo acciones de activación/desactivación y consultas por bloque.*

**Anexo 6 – Figura 7**

**Procedimiento para generar y gestionar reservas**

El siguiente anexo documenta el procedimiento para la creación y seguimiento de reservas dentro del sistema, incluyendo rutas, opciones de interfaz, pasos y buenas prácticas:

***Tabla 7: Gestión de Reservas***

|  |  |  |  |
| --- | --- | --- | --- |
| **Acción** | **Ruta / Ubicación** | **Descripción** | **Observaciones / Buenas prácticas** |
| Crear reserva vía formulario | reservas/create | Permite registrar una reserva directamente completando los campos solicitados, como fecha, bloque, aula, productos y observaciones. | Usar información precisa sobre la disponibilidad de aulas y productos; revisar conflictos de horario antes de enviar. |
| Crear reserva vía calendario | reservas/calendario | Permite seleccionar visualmente la fecha y el bloque horario, facilitando la planificación de reservas. | Garantizar que los bloques y aulas estén previamente configurados y activos. |
| Consultar mis reservas | mis-reservas | Permite al usuario revisar las reservas creadas por él, incluyendo estado y detalles de cada una. | Mantener actualizado el seguimiento de reservas propias para control personal. |
| Reservas pendientes de aprobación | reservas/pendientes-aprobacion | Permite al administrador o encargado gestionar las reservas que requieren autorización. | Facilita la revisión rápida de solicitudes y evita conflictos de disponibilidad. |
| Ver detalle/historial de reservas | reservas/{reserva}, reservas/{reserva}/historial | Permite consultar información detallada de una reserva específica, así como su historial de modificaciones y cambios de estado. | Útil para auditorías internas y seguimiento de cambios en la reserva. |

******

*Figura 7. Interfaz y rutas utilizadas para la creación, seguimiento y gestión de reservas en el sistema, mostrando opciones de formulario y calendario, así como consultas de historial y estado.*