

UNIVERSIDAD NACIONAL DEL ALTIPLANO

FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA



PROJECT SCRUM METHODOLOGY

**Digital Platform for the Valuation and
Commercialization of Alpaca Fiber to
Empower Producers and Foster Business
Success**

Luz Bella Valenzuela Narvaez

Teacher: Torres Cruz Fred

Course: Software Engineering

Semester: VII

Puno-Perú

2024

Project

Digital Platform for the Valuation and Commercialization of Alpaca Fiber to Empower Producers and Foster Business Success

Project Management

Scrum Agile Methodology

Scrum will be adopted due to its efficiency in managing projects through short and focused iterations, essential for AlpacaConnect's development. Scrum will facilitate the incremental delivery of features, ensuring the platform continuously evolves to meet user needs and enhance their direct marketing experience.

Project Objective

The objective of AlpacaConnect is to improve the economic situation of alpaca fiber producers by providing them with a comprehensive digital platform for direct valorization and commercialization. It will facilitate direct market access, ensure product quality, empower producers with production tracking tools and business training, promote producer autonomy, and foster transparency and economic development in alpaca communities.

Functional Requirements

Producers

1. Producer Registration: Allow producers to register on the platform by providing basic information and details about their products.
2. Product Catalog Management: Enable producers to add, edit, and delete products from their catalog, including descriptions, prices, and photos.
3. Production Tracking: Provide tools for producers to record and track the entire production process, from camelid breeding to the final product.
4. Statistical Visualization: Allow producers to access statistics on their product sales and market trends.

Buyers

1. Buyer Registration: Allow buyers to register on the platform by providing basic information.
2. Product Exploration: Enable buyers to search and explore alpaca fiber products in the catalog.
3. Shopping Cart: Allow buyers to add products to the shopping cart and complete transactions securely.

4. Product Details Viewing: Provide detailed information about products (from breeding to the final product), including prices and photos.
5. Ratings and Comments: Allow buyers to give ratings and comments on products.
6. Order Tracking: Allow buyers to track the status of their orders and receive notifications about the shipping process.

Administrators

1. User Management: Allow administrators to manage user accounts, including producers and buyers.
2. Content Moderation: Moderate and manage platform content, including products and comments.
3. Data Analysis: Provide tools for analyzing sales data, market trends, and user preferences.
4. System Maintenance: Perform system maintenance and updates to ensure optimal operation.
5. Customer Support: Provide customer support to resolve queries and issues reported by users.

Non-functional Requirements

1. **Security:** Implement robust authentication, encrypted communication, and secure data storage.
2. **Reliability:** Operate reliably under normal conditions, with minimal downtime or errors.
3. **Maintainability:** Facilitate maintenance and updates through well-structured code, clear documentation, and modular architecture.
4. **Scalability:** Be scalable to accommodate an increase in the number of users and data volume without significant performance degradation.
5. **Portability:** Be compatible with various operating systems, web browsers, and devices.
6. **Usability:** Be user-friendly with intuitive interfaces and clear navigation.
7. **Performance:** Operate efficiently with fast response times and minimal latency.
8. **Authentication and Authorization:** Enforce user authentication and authorization mechanisms to control access to sensitive functions and data.
9. **Data Integrity:** Ensure data integrity through measures to prevent unauthorized access, modification, or corruption of data.
10. **Compliance:** Comply with relevant regulations and standards, including those related to data protection, industry security, and accessibility.

Architecture Design

Microservices Architecture

Microservices architectures offer an updated, scalable, and distributed system. This means that the system can easily expand according to needs and be distributed into multiple independent components. It is compatible with PHP and Apache in XAMPP.

Each microservice can be developed and deployed independently, and Bootstrap can be used to design user interfaces.

Features

- Decoupling of components: Allows for the development, deployment, and operation of services independently, streamlining application management and minimizing impacts on other services in case of changes or updates.
- Ease of maintenance and testing: Facilitates experimentation with new features, their rollback if they don't work, and efficient identification and correction of errors, improving software development agility and quality.
- Small and agile teams: Encourages autonomous and multidisciplinary teams that can work independently, reducing bureaucracy and accelerating development processes.
- Organization around business capabilities: Teams focus on specific functionalities, optimizing efficiency and allowing better alignment with business needs.
- Infrastructure automation: Improves efficiency in development and deployment by allowing for automatic creation and deployment of services, ensuring fast and consistent delivery.

Microservices Architecture for Alpaca Fiber Platform

General Description:

The microservices architecture divides the platform into several independent services, each responsible for specific functionality. These microservices communicate with each other via RESTful APIs. The platform will be developed in PHP with Visual Studio, deployed in a XAMPP environment (Apache, MySQL), and Bootstrap will be used for the frontend.

Microservices Architecture Components

1. User Management Service

- **Functionality:** Registration, authentication, profile management.
- **Technologies:** PHP, MySQL.
- **Endpoints:**
 - POST /register
 - POST /login
 - GET /profile

- PUT /profile

2. Product Management Service

- **Functionality:** Add, edit, delete alpaca fiber products.
- **Technologies:** PHP, MySQL.
- **Endpoints:**
 - POST /products
 - GET /products
 - PUT /products/{id}
 - DELETE /products/{id}

3. Sales Management Service

- **Functionality:** Purchase processing, sales statistics.
- **Technologies:** PHP, MySQL.
- **Endpoints:**
 - POST /sales
 - GET /sales
 - GET /sales/stats

4. Customer Support Service

- **Functionality:** Management of queries and customer support.
- **Technologies:** PHP, MySQL.
- **Endpoints:**
 - POST /support
 - GET /support/tickets
 - PUT /support/tickets/{id}
 - DELETE /support/tickets/{id}

5. Advertising Service

- **Functionality:** Management and display of advertising related to alpaca fiber.
- **Technologies:** PHP, MySQL.
- **Endpoints:**
 - POST /ads
 - GET /ads
 - PUT /ads/{id}
 - DELETE /ads/{id}

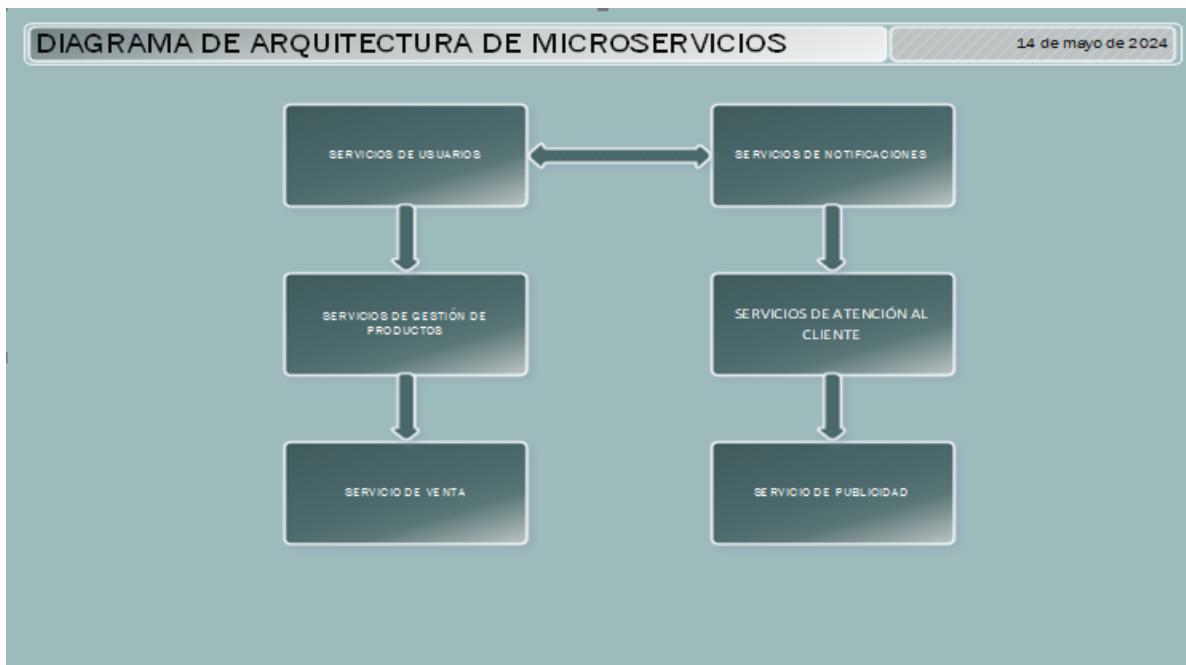
6. Notification Service

- **Functionality:** Sending notifications to users about their products and purchases.

- **Technologies:** PHP, MySQL.

- **Endpoints:**

- POST /notifications
- GET /notifications



(a) Microservices Architecture Diagram

Technologies and tools

Programming languages

- **PHP:** Main language used for server-side development, handling business logic and database interaction.
- **JavaScript:** Used for client-side programming, providing interactivity and dynamism to the user interface.

Frameworks and Libraries

- **Bootstrap:** CSS framework used to design a responsive and modern user interface.
- **Laravel:** PHP framework (optional if chosen) that facilitates web application development through an MVC structure and integrated tools for common tasks.

Web server

- **Apache:** Web server used to host the application, included in the XAMPP distribution.

Development Environments

- **XAMPP:** Software package that includes Apache, MySQL, PHP, and Perl, facilitating the setup of the local development environment.
- **Visual Studio Code:** Source code editor used to write, debug, and maintain project code.

Databases

- **MySQL:** Relational database management system used to store application data, included in XAMPP.

Version control

- **Git:** Distributed version control system used to manage source code history and collaborate with other developers.
- **GitHub:** Web-based platform for hosting Git repositories and facilitating collaboration in project development.

Automation Tools

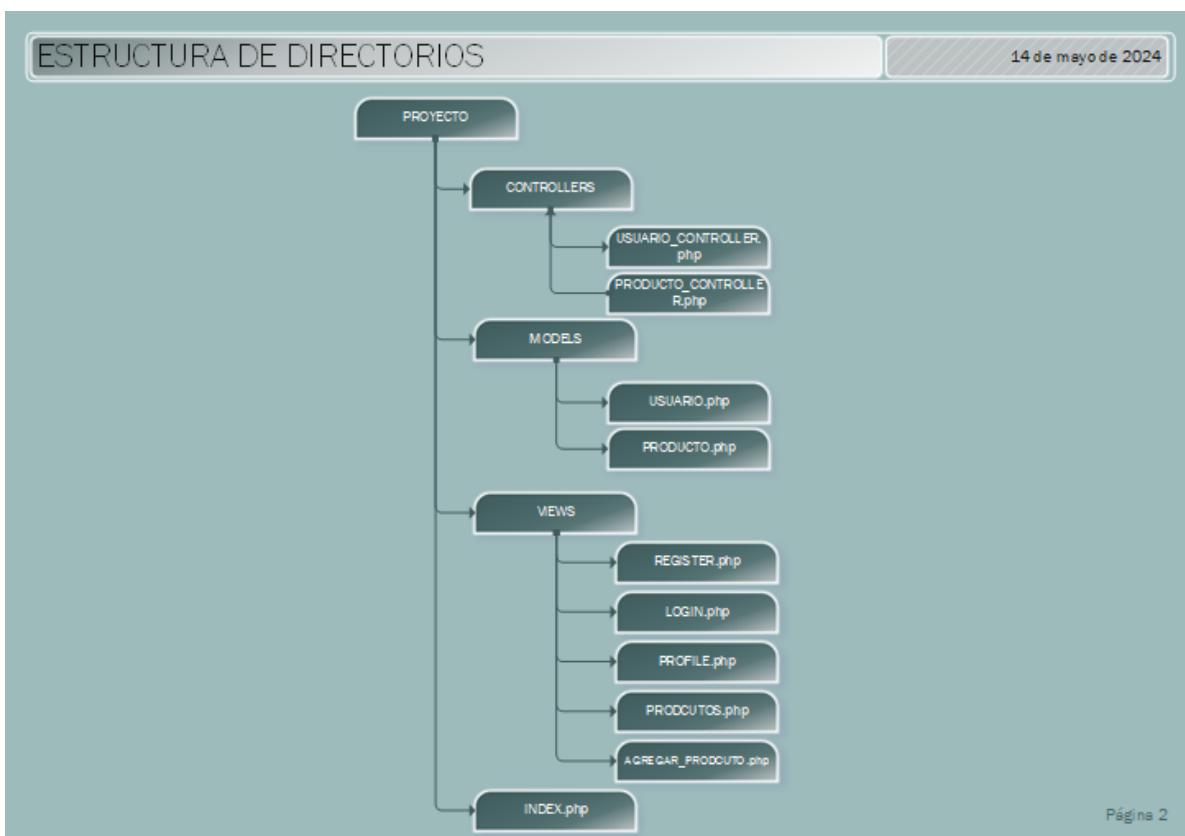
- **Composer:** Dependency manager for PHP, used to install and manage packages and libraries.
- **npm (Node Package Manager):** Used to manage JavaScript dependencies and frontend build tools.

Development Methodologies

- **Agile:** Methodologies adopted to promote collaboration, continuous integration (CI), continuous delivery (CD), and continuous deployment (also CD), improving the efficiency and quality of development.

Security

- **HTTPS:** Security protocol used to secure communication between the client and server.
- **Implementation of security measures:** Authentication, authorization, and data encryption to protect users' sensitive information.



(a) Directory structure

Security for AlpacaConnect

Security Practices

- **Authentication:** Implement secure authentication using JWT tokens to handle user sessions. Each user must authenticate before accessing any protected resources on the platform.
- **Authorization:** Define clear roles and permissions (e.g., producers and buyers) and use access control policies to ensure that users can only access resources they are allowed to.
- **Data Encryption:** Ensure that all sensitive data, such as passwords and payment details, is encrypted both in transit (using HTTPS) and at rest (using database encryption techniques).
- **API security:** Implement security measures in the API, such as input validation, rate limiting, and protection against common attacks (e.g., SQL injection, XSS).

Security Tests

- **Vulnerability scan:** Conduct regular vulnerability scans using automated scanning tools (e.g., OWASP ZAP) to identify and mitigate potential security weaknesses in the code and infrastructure.

- **Penetration tests:** Hire security experts or use penetration testing services to simulate real attacks and discover vulnerabilities that could be exploited by attackers. These tests should be performed periodically and after significant changes to the system.

AlpacaConnect Maintenance

Corrective maintenance

- **Error detection and resolution:** Establish an error tracking system to detect and fix any issues on the platform quickly. Use tools like Sentry to monitor and log errors in real-time.
- **User support:** Implement a technical support channel for users to report issues and receive timely assistance. This may include a ticketing system and live chat.

Preventive Maintenance

- **Software updates:** Keep the Apache server, MySQL database, and other platform components (PHP, XAMPP, etc.) up to date to avoid known vulnerabilities and improve performance.
- **Proactive Monitoring:** Use monitoring tools like Nagios or New Relic to monitor system health, server performance, and database activity, identifying potential issues before they affect users.

Evolutionary Maintenance

- **Implementation of new functionalities:** Plan and develop new features and enhancements on the platform, such as integration with new payment methods, multilingual support, and UI improvements.
- **Continuous optimization:** Continuously review and improve code and database queries to ensure optimal performance. Conduct regular code reviews and optimize SQL queries to enhance speed and efficiency.

Adaptive Maintenance

- **Adaptation to new technologies:** Stay updated on new technologies and trends in web development, and adapt the platform to leverage these innovations. This may include using new versions of frameworks and libraries.
- **Normative compliance:** Ensure that the platform complies with local and international regulations.

Scrum Methodology with JIRA Platform

Producers

User Registration

- Design the database to store user information.
- Implement the user registration interface.
- Develop backend logic for registration and data storage.
- Implement a password encryption function.
- Implement login functionality.
- User testing and bug fixes.

Platform Access

- Create a registration form.
- Create a login form.
- Create data retrieval functionality for registrations.

Product Catalog Management

- Design the database to store products.
- Create a user interface to add, edit, and delete products.
- Develop backend logic for product management.
- Add functionality to upload images.
- User testing and bug fixes.

Production Tracking

- Design the database for production tracking.
- Create the user interface to record the production process.
- Develop backend logic to manage production tracking.
- Integration testing and bug fixes.

Statistics Visualization

- Design the database to store and query statistics.
- Create the user interface to display statistics.
- Implement backend logic to calculate and return statistics.
- User testing and bug fixes.

Buyers

Product Exploration

- Design the user interface for product exploration.
- Implement search and filter logic for products.
- User testing and bug fixes.

Shopping Cart

- Design the database for the shopping cart.
- Implement the user interface for the shopping cart.
- Develop backend logic to manage the cart and transactions.
- Implement secure payment methods (Yape QR).
- User testing and bug fixes.

Product Detail View

- Create the user interface for product detail view.
- Implement backend logic to provide detailed information.
- User testing and bug fixes.

Ratings and Reviews

- Design the database to store ratings and reviews.
- Create the user interface to add and view ratings and reviews.
- Implement backend logic to manage ratings and reviews.
- User testing and bug fixes.

Order Tracking

- Design the database to store order statuses.
- Implement the user interface for order tracking.
- Develop backend logic to update and display order status.
- User testing and bug fixes.

Administrators

User Management

- Design the database to manage users.
- Implement the user interface for user management.
- Develop backend logic to manage user accounts.
- Integration testing and bug fixes.

Content Moderation

- Create the user interface for content moderation.
- Implement backend logic to manage products and comments.
- User testing and bug fixes.

Data Analysis

- Design the database to store and analyze data (Hash).
- Create the user interface for data analysis visualization.
- Implement backend logic to calculate and display analysis.
- User testing and bug fixes.

System Maintenance

- Plan and execute system updates.
- Perform regression testing after updates.
- Implement monitoring processes to ensure optimal performance.
- Document maintenance procedures.

Customer Support

- Implement a ticket system for customer support.
- Create a user interface for ticket tracking and management.
- Develop backend logic to manage support requests.
- User testing and bug fixes.

This link is the Jira platform

Jira platform

	Tipo	# Clave	Resumen	Estado	Sprint	Persona asignada	Fe
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-1	Registro de los usuarios	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-3	Diseñar la base de datos para almacenar la información de lo...	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-4	Implementar la interfaz de registro de usuarios	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-5	Desarrollar la lógica de backend para el registro y almacena...	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-6	Implementar una función de encriptación para las contraseñas	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-7	Implementar el inicio de sesión	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-8	Pruebas de usuario y corrección de errores	TAREAS POR HAC...	Tablero Sprint 1		

(a) PPD-1

	Tipo	# Clave	Resumen	Estado	Sprint	Persona asignada	Fe
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-2	Acceso a la plataforma digital	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-9	Crear un formulario de registro	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-10	Crear un formulario de acceso	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-11	Crear la función de recepción de datos de registros	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-12	PRODUCTORES_Gestión de catálogo de productos	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-25	Diseñar la base de datos para almacenar productos.	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-26	Crear interfaz de usuario para agregar, editar y eliminar prod...	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-27	Implementar la lógica de backend para la gestión de product...	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-28	Agregar funcionalidad para subir imágenes.	TAREAS POR HAC...	Tablero Sprint 1	Sin asignar	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PPD-29	Pruebas de usuario y corrección de errores.	TAREAS POR HAC...	Tablero Sprint 1		

(a) PDD-2 y PDD-3

	Tipo	# Clave	Resumen	Estado	Sprint	Persona asignada
□	▼	PPD-13	Seguimiento de producción	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-30	Diseñar la base de datos para el seguimiento de producción.	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-31	Crear la interfaz de usuario para registrar el proceso de prod...	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-32	Implementar la lógica de backend para gestionar el seguimie...	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-33	Pruebas de integración y corrección de errores.	TAREAS POR HAC...	Tablero Sprint 1	
□	▼	PPD-14	Visualización de estadísticas	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-34	Diseñar la base de datos para almacenar y filtro de los datos.	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-35	Crear la interfaz de usuario para mostrar estadísticas de las v...	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-36	Implementar la lógica de backend para calcular y devolver es...	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-37	Pruebas de usuario y corrección de errores.	TAREAS POR HAC...	Tablero Sprint 1	

(a) PDD-4 y PDD-5

	Tipo	# Clave	Resumen	Estado	Sprint	Persona asignada
□	▼	PPD-15	CLIENTES_Exploración de productos	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-38	Diseñar la interfaz de usuario para la exploración de product...	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-39	Implementar la lógica de búsqueda y filtrado de productos.	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-40	Pruebas de usuario y corrección de errores.	TAREAS POR HAC...	Tablero Sprint 1	
□	▼	PPD-16	Carrito de compras	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-41	Diseñar la base de datos para el carrito de compras.	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-42	Implementar la interfaz de usuario para el carrito de compras.	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-43	Desarrollar la lógica de backend para gestionar el carrito y la...	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-44	Implementar métodos de pago seguros.(Yape QRL)	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-45	Pruebas de usuario y corrección de errores.	TAREAS POR HAC...	Tablero Sprint 1	

(a) PDD-6 y PDD-7

	Tipo	# Clave	Resumen	Estado	Sprint	Persona asignada
□	▼	PPD-17	Visualización de detalles del producto	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-46	Crear la interfaz de usuario para la visualización de detalles d...	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-47	Implementar la lógica de backend para proporcionar informa...	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-48	Pruebas de usuario y corrección de errores.	TAREAS POR HAC...	Tablero Sprint 1	
□	▼	PPD-18	Valoraciones y comentarios	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-49	Diseñar la base de datos para almacenar valoraciones y com...	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-50	Crear la interfaz de usuario para agregar y ver valoraciones y ...	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-51	Implementar la lógica de backend para gestionar valoracione...	TAREAS POR HAC...	Tablero Sprint 1	
□		PPD-52	Pruebas de usuario y corrección de errores.	TAREAS POR HAC...	Tablero Sprint 1	

(a) PDD-8 y PDD-9

<input type="checkbox"/>	Tipo	# Clave	Resumen	⊕ Estado	⌚ Sprint	@ Persona...
<input type="checkbox"/>	+	PPD-19	Seguimiento de pedidos	TAREAS POR HAC...	Tablero Sprint 1	
+ <input type="checkbox"/>	PPD-53	Diseñar la base de datos para almacenar el estado de los pe...	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-54	Implementar la interfaz de usuario para el seguimiento de pe...	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-55	Desarrollar la lógica de backend para actualizar y mostrar el ...	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-20	ADMINISTRADORES_Gestión de usuarios	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-56	Diseñar la base de datos para gestionar usuarios.	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-57	Implementar la interfaz de usuario para la gestión de usuarios.	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-58	Desarrollar la lógica de backend para gestionar cuentas de u...	TAREAS POR HAC...	Tablero Sprint 1		

(a) PDD-10 y PDD-11

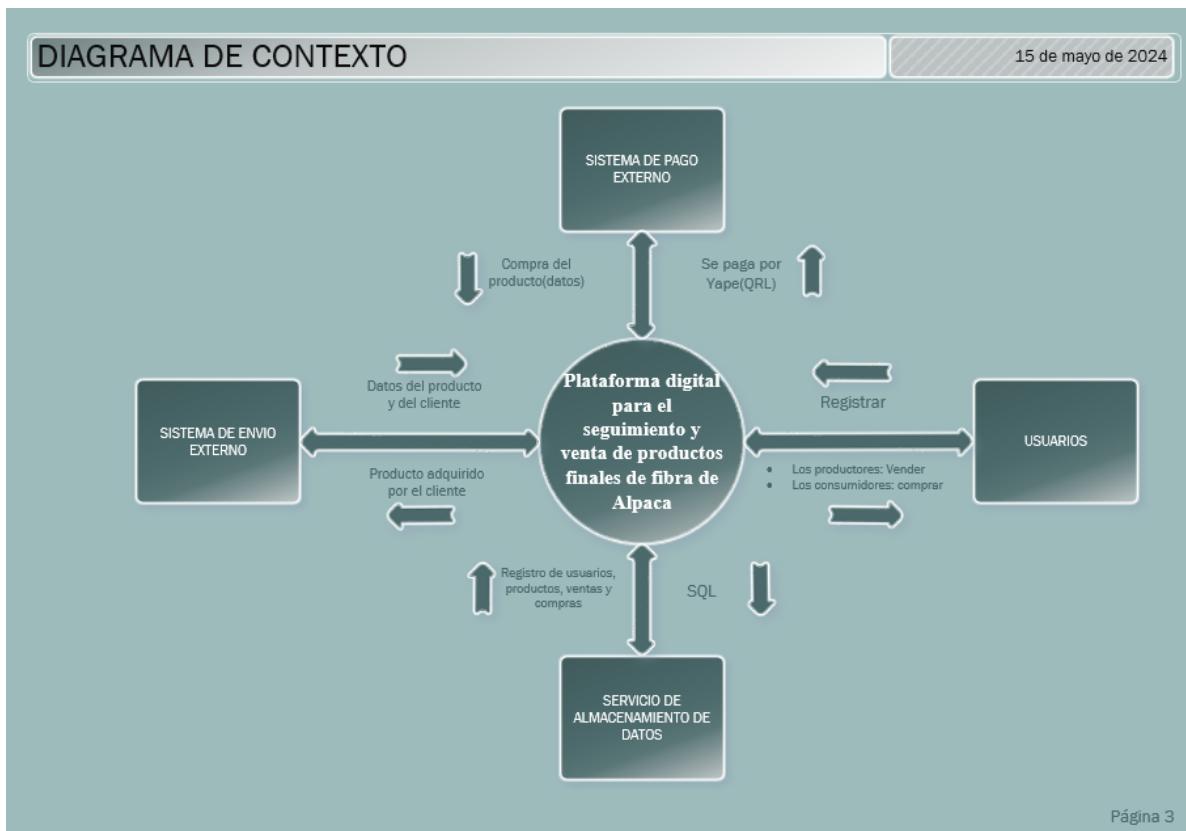
<input type="checkbox"/>	PPD-21	Moderación de contenido	TAREAS POR HAC...	Tablero Sprint 1	
<input type="checkbox"/>	PPD-59	Crear la interfaz de usuario para la moderación de contenido.	TAREAS POR HAC...	Tablero Sprint 1	
<input type="checkbox"/>	PPD-60	Implementar la lógica de backend para gestionar productos ...	TAREAS POR HAC...	Tablero Sprint 1	
<input type="checkbox"/>	PPD-61	Pruebas de usuario y corrección de errores.	TAREAS POR HAC...	Tablero Sprint 1	
<input type="checkbox"/>	PPD-22	Análisis de datos	TAREAS POR HAC...	Tablero Sprint 1	
<input type="checkbox"/>	PPD-62	• Diseñar la base de datos para almacenar y analizar datos.(H...	TAREAS POR HAC...	Tablero Sprint 1	
<input type="checkbox"/>	PPD-63	Crear la interfaz de usuario para la visualización de análisis d...	TAREAS POR HAC...	Tablero Sprint 1	
<input type="checkbox"/>	PPD-64	Implementar la lógica de backend para calcular y mostrar an...	TAREAS POR HAC...	Tablero Sprint 1	

(a) PDD-12 y PDD-13

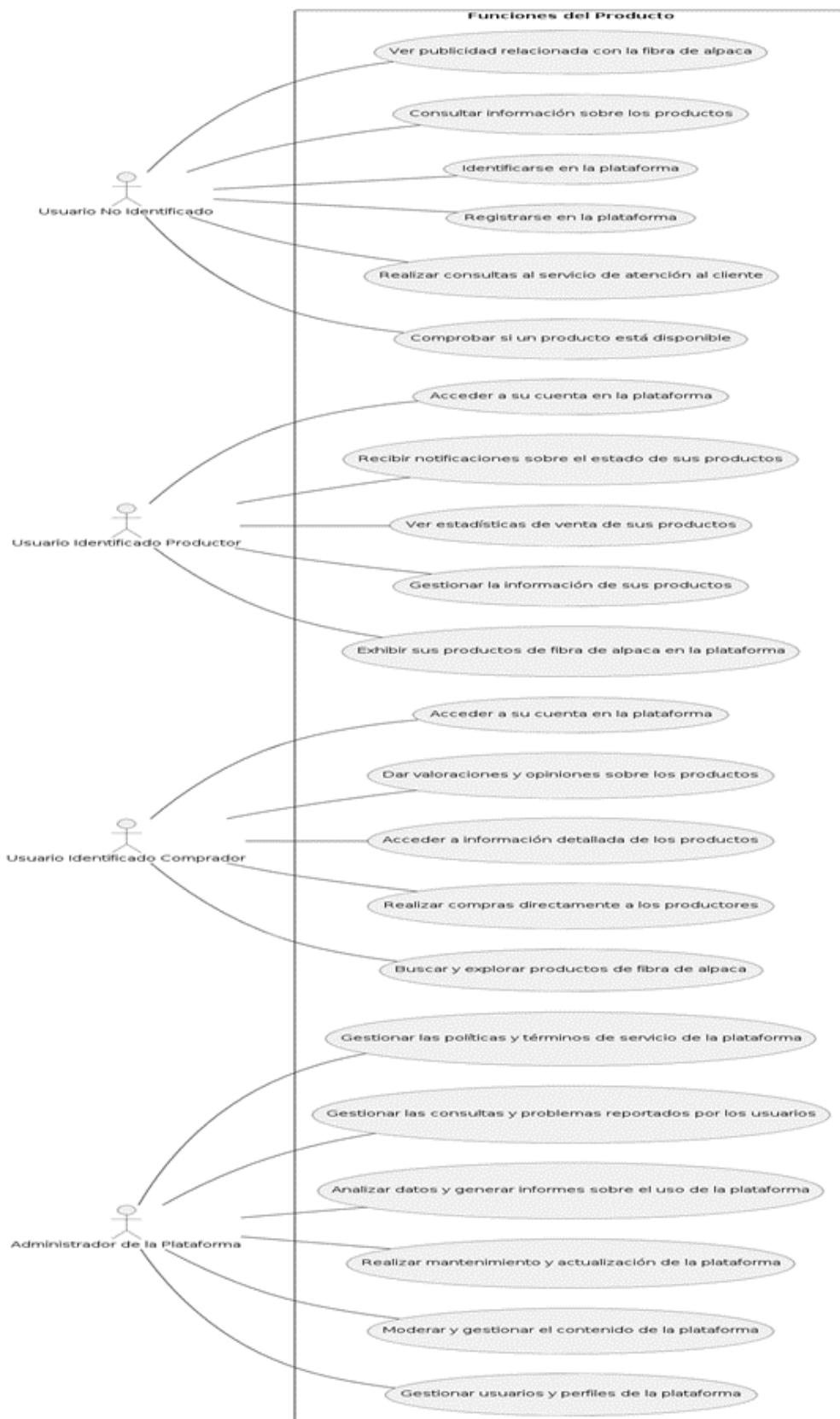
<input type="checkbox"/>	Tipo	# Clave	Resumen	⊕ Estado	⌚ Sprint	@ Persona asignada
<input type="checkbox"/>	PPD-23	Mantenimiento del sistema	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-65	Planificar y ejecutar actualizaciones del sistema.	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-66	Realizar pruebas de regresión después de las actualizaciones.	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-67	Implementar procesos de monitoreo para garantizar el funcione...	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-24	Soporte al cliente	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-68	Implementar un sistema de tickets para soporte al cliente.	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-69	Crear una interfaz de usuario para el seguimiento y gestión d...	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-70	Desarrollar la lógica de backend para gestionar solicitudes d...	TAREAS POR HAC...	Tablero Sprint 1		
<input type="checkbox"/>	PPD-71	Pruebas de usuario y corrección de errores.	TAREAS POR HAC...	Tablero Sprint 1		

(a) PDD-14 y PDD-15

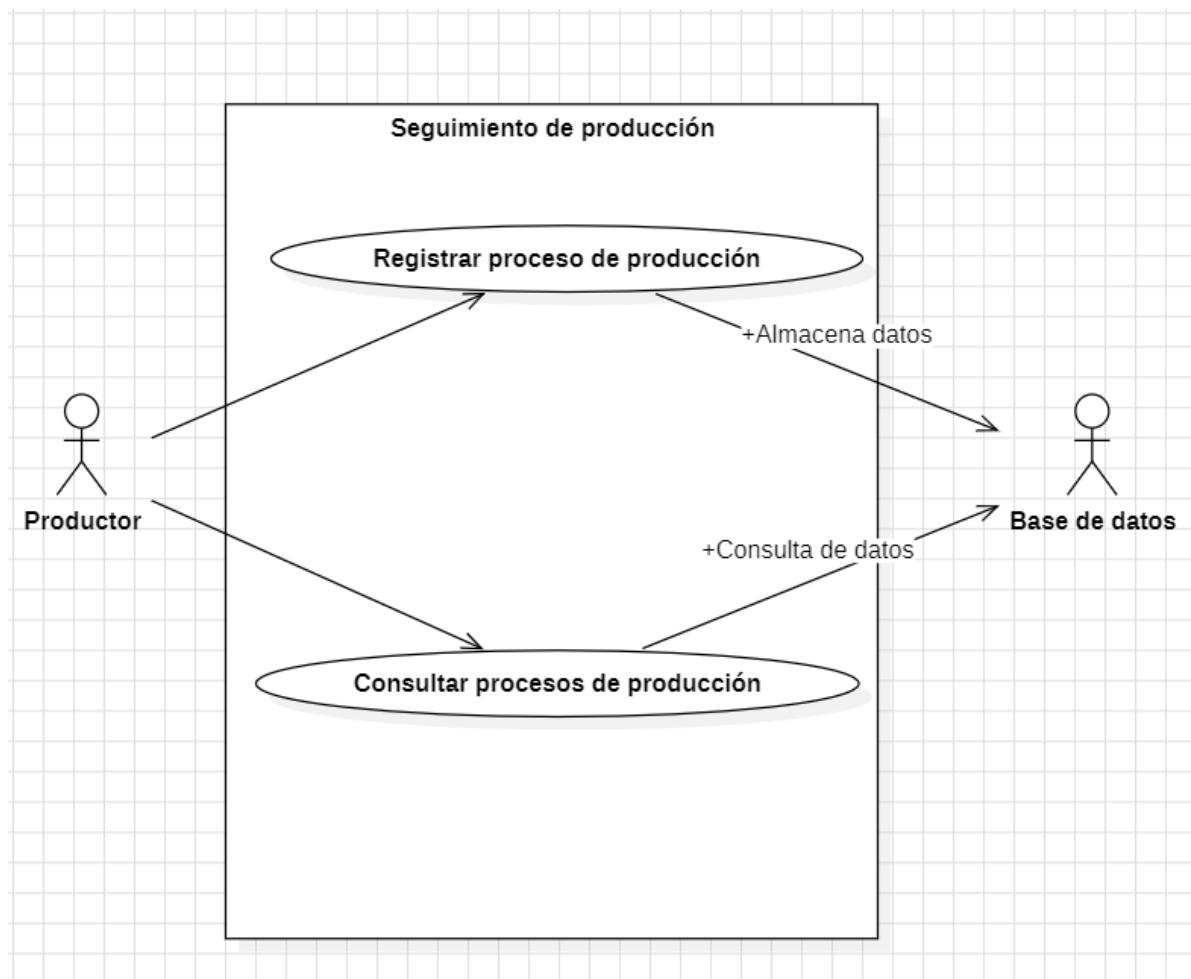
Project diagrams



(a) Context diagram

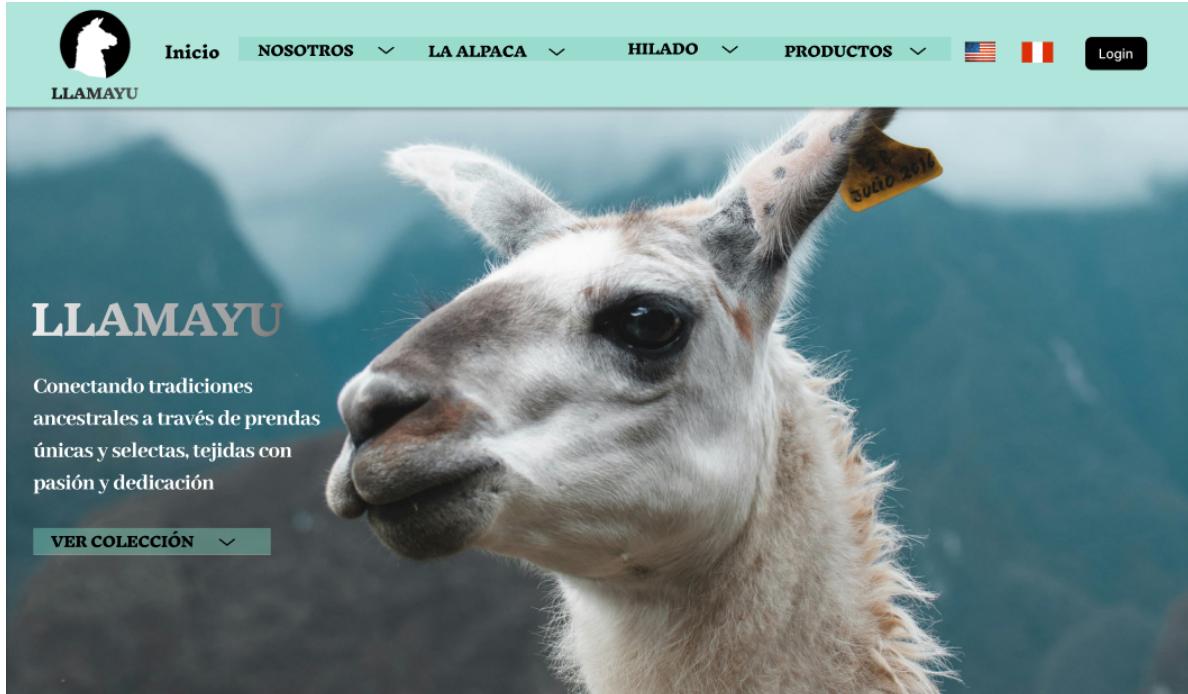


(a) Use case diagram

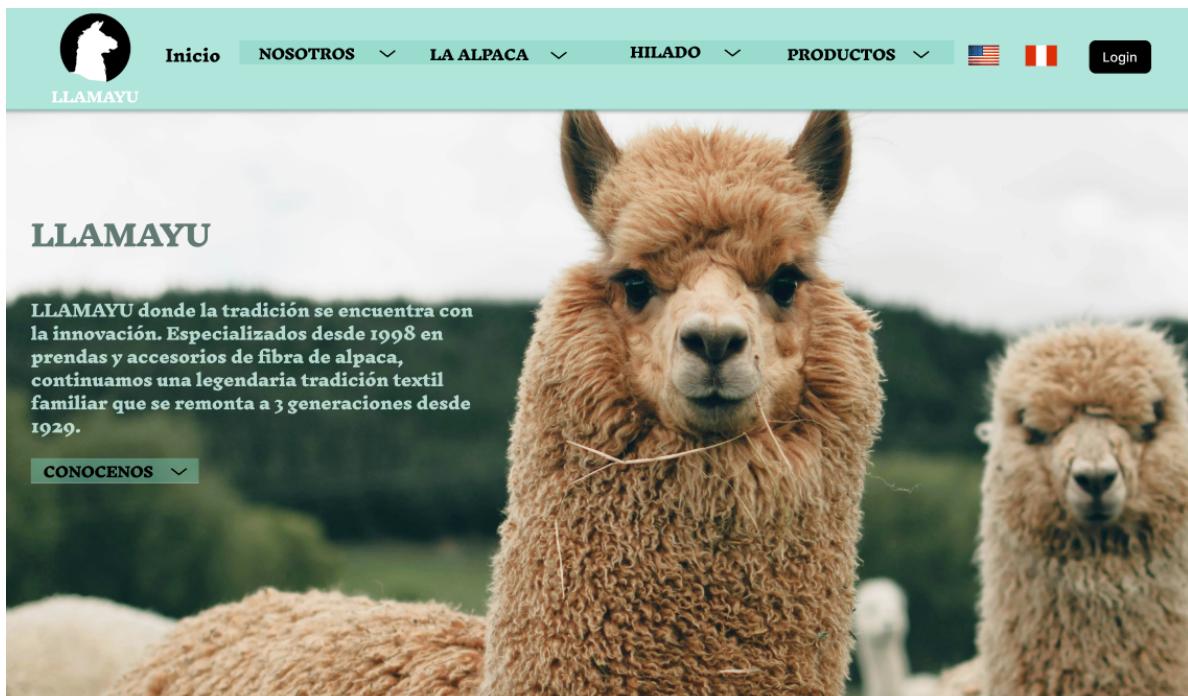


(a) Production Tracking Use Case Diagram

Digital platform previews



(a) Digital platform previews



(a) Digital platform previews



(a) Digital platform previews