

Práctica 1

Teniendo en cuenta los nuevos conceptos y habilidades aprendidas del proceso unificado, realiza las mejoras que considere necesario del trabajo práctico del Ejercicio 8.

1. Análisis de Requerimientos:

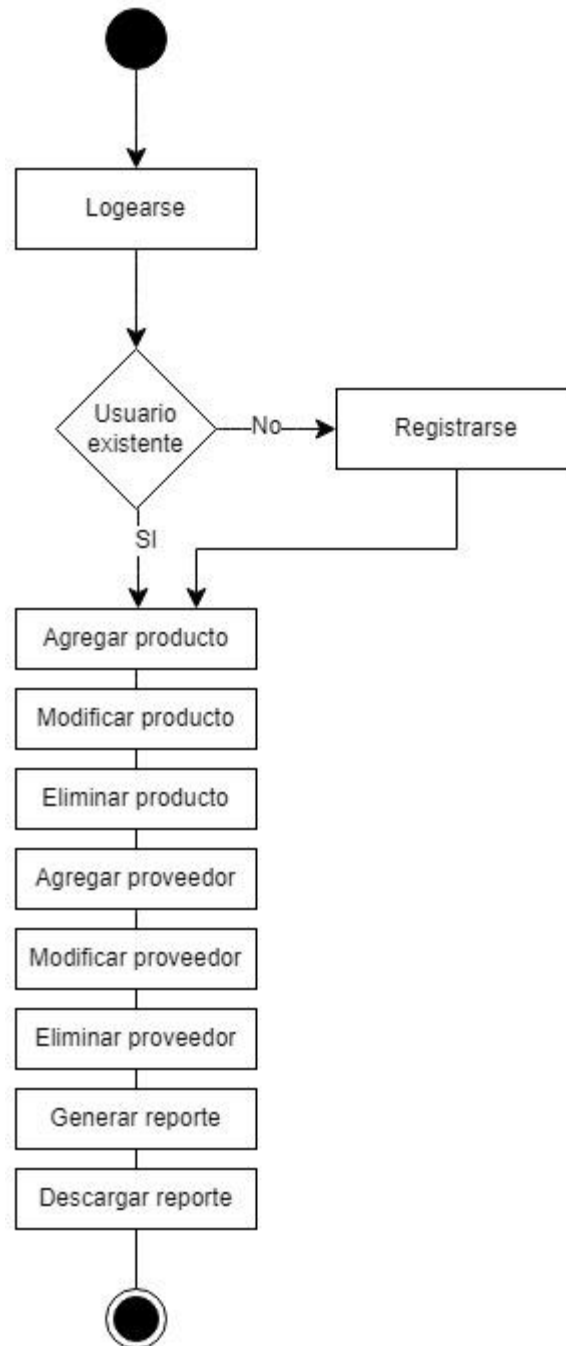
- **Ejercicio 1:** Un cliente te solicita una aplicación web para gestionar su inventario. Define los requisitos funcionales y no funcionales del sistema.
 - RF1: El usuario debe poder crear, modificar y eliminar productos.
 - RF2: El usuario debe poder agregar del producto datos como: nombre, código, descripción, stock, proveedor.
 - RF3: El usuario debe poder crear, modificar y eliminar información de proveedores, como nombre, contacto, dirección y productos.
 - RF4: El usuario debe poder generar y descargar reportes de inventario por producto, proveedor y categoría.
 - RF5: El usuario debe recibir notificaciones cuando el stock llega a un valor mínimo.
 - RN6: La aplicación debe poder ser vista adecuadamente desde computadoras y celulares.
 - RN7: Las páginas deben tener buena performance.
 - RN8: Los datos deben poderse encriptar.
 - RN9: La aplicación debe poder procesar grandes volúmenes de datos.
- **Ejercicio 2:** Redacta un caso de uso para la funcionalidad de "Agregar un nuevo producto" en la aplicación web del ejercicio 1.

| | Agregar producto | Fecha: 09/07/2024 |
|---------------|--|----------------------|
| Código | CUD01 | |
| Nombre | Agregar producto | |
| Referencias | RF1 | |
| Autor | Contrera Valentina | |
| Revisor | Sartorio Alejandro | |
| Descripción | Este caso de uso permite agregar un nuevo producto especificando los detalles necesarios | |
| Actores | Usuario | |
| Pre condición | El usuario debe estar logueado en la aplicación El usuario debe tener permiso de agregar producto | |
| Condición | El usuario presiona el botón "Agregar producto" | |

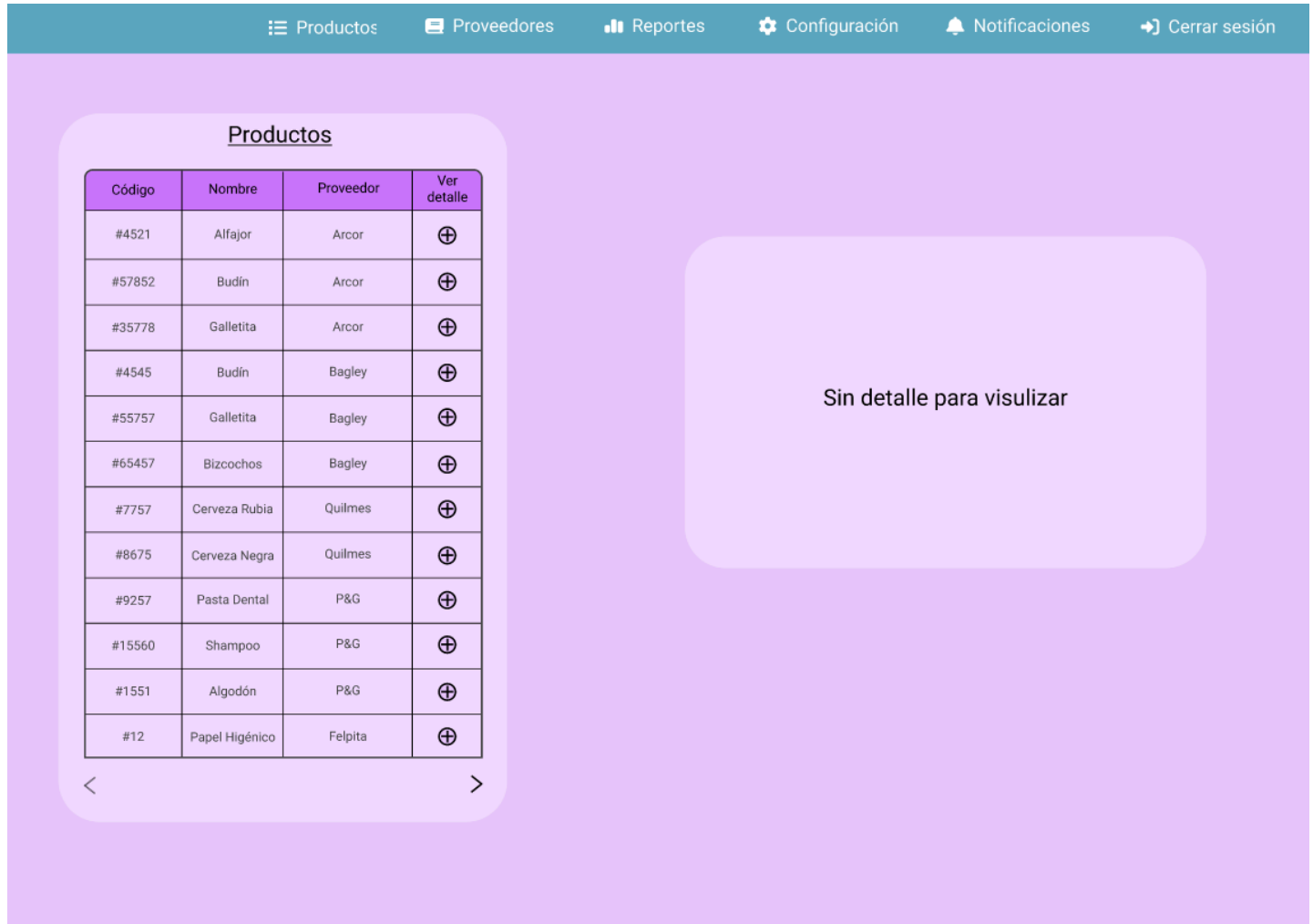
| Curso Básico | Acción |
|-------------------|--|
| 1 | El usuario presiona el botón "Agregar producto" |
| 2 | El sistema redirige a una pantalla, con un formulario con dos botones "Cancelar" y "Guardar", este último deshabilitado, y los datos a completar son: nombre, código, descripción, stock, proveedor. |
| 3 | El usuario completa los datos |
| 4 | El sistema verifica los datos y habilita el botón "Guardar" |
| 5 | El usuario presiona el botón "Guardar" |
| 6 | El sistema guarda el producto, muestra cartel específico y vuelve atrás |
| Curso Alternativo | Acción |
| 3.a | El usuario decide no crear un producto, aprieta el botón "Cancelar" y el sistema vuelve atrás |
| Post condición | Producto guardado con éxito |

2. Diseño del Sistema:

- **Ejercicio 3:** Elabora un diagrama de flujo de datos para la aplicación web del ejercicio 1.



- **Ejercicio 4:** Diseña la interfaz de usuario para la pantalla de "Inicio" de la aplicación web del ejercicio 1.



3. Diseño del Programa:

- **Ejercicio 5:** Elige una arquitectura adecuada para la aplicación web del ejercicio 1 y justifica tu elección.

Una arquitectura en capas podría ser mejor. Capa de Presentación: Responsable de la interfaz de usuario y la interacción con el usuario; Capa de Aplicación: Contiene la lógica de negocio y la coordinación entre las diferentes funcionalidades; Capa de Dominio: Representa el modelo de datos y las reglas de negocio; Capa de Infraestructura: Maneja la persistencia de datos, la seguridad, las comunicaciones, etc. Esta arquitectura permite una mejor separación de responsabilidades, lo cual facilita el mantenimiento y la evolución de la aplicación a lo largo del tiempo. Además, se alinea mejor con los principios del Proceso Unificado, como la separación de intereses y la arquitectura dirigida por los casos de uso.

- **Ejercicio 6:** Diseña la base de datos para la aplicación web del ejercicio 1.
 - Entidad producto:

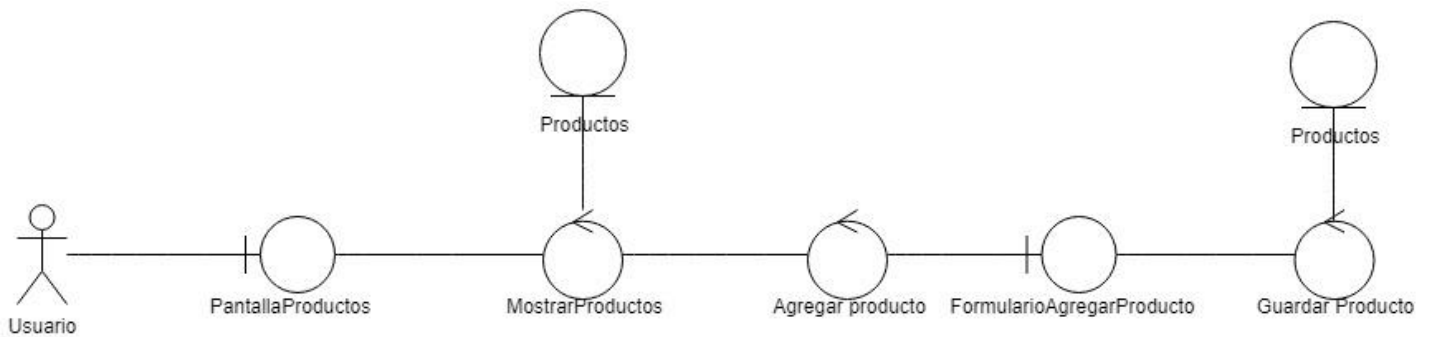
- ProductoID
- Nombre
- Código
- Descripción
- Cantidad
- ProveedorID
- Categoría
- Entidad proveedor:
 - ProveedorID
 - Nombre
 - Contacto
 - Dirección
- Entidad movimiento_inventario:
 - MovimientoID
 - ProductoID
 - Cantidad
 - Tipo (Entrada/Salida)
 - Fecha
 - UsuarioID
- Entidad usuario:
 - UsuarioID
 - Nombre
 - Apellido
 - Correo
 - Contraseña
- Entidad notificacion:
 - NotificacionID
 - Mensaje
 - Tipo (Entrada/Salida)
 - Fecha
 - ProductoID
 - UsuarioID
- Entidad categoria:
 - CategoricalID
 - Nombre
 - Descripcion

4. Diseño:

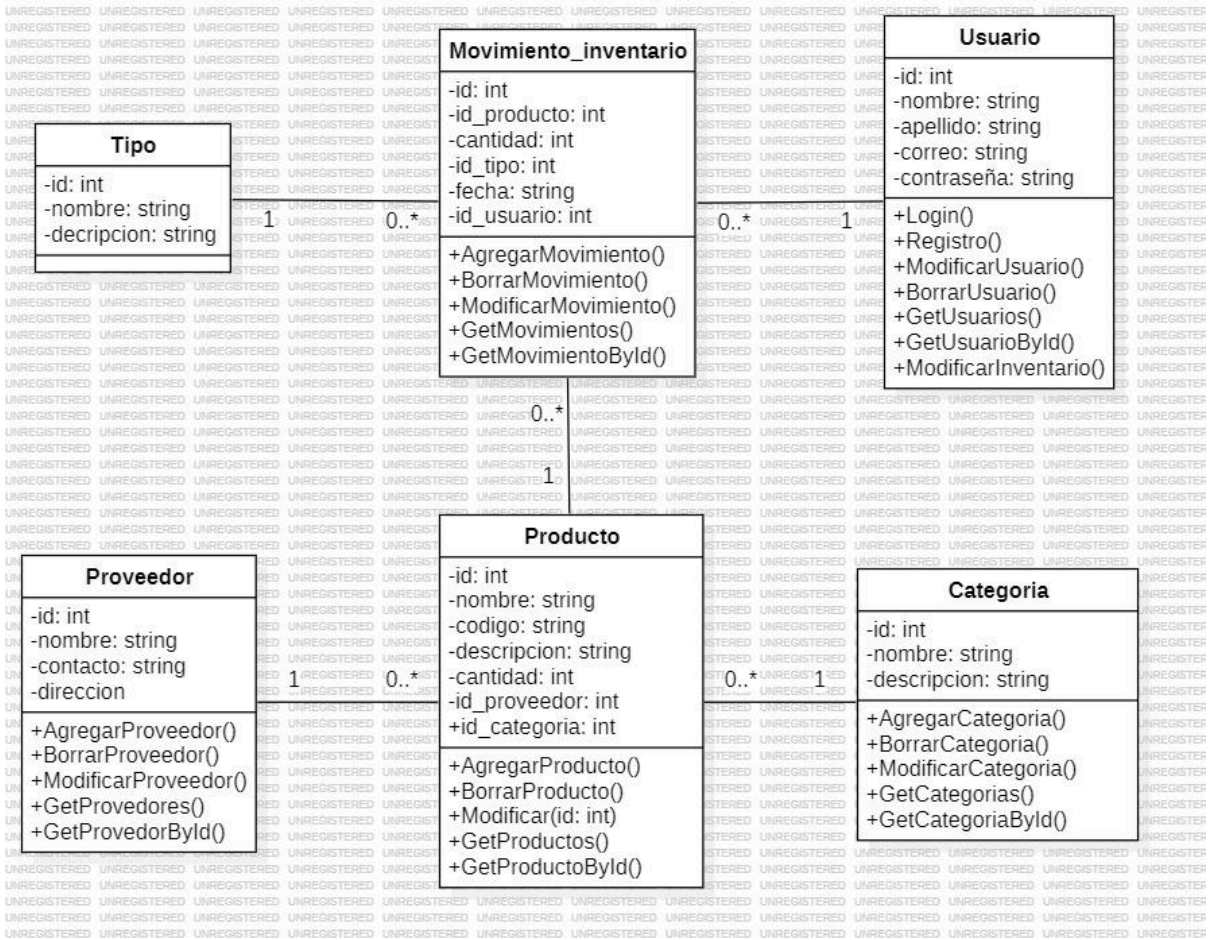
Utilizando los siguientes diagrama resuelva los casos de usos de los ejercicios 7 y 8:

1. **Diagrama de Dominio:** Identifica las entidades, atributos y relaciones del sistema.
2. **Diagrama de Robustez:** Analiza cómo el sistema responde a diferentes escenarios de uso.
3. **Prototipo:** Crea una versión simplificada del sistema para probar la usabilidad y funcionalidad.
4. **Diagrama de Secuencia:** Describe la interacción entre los diferentes objetos del sistema.
5. **Diagrama de Clases:** Define las clases, sus atributos, métodos y relaciones

- **Ejercicio 7:** Implementa la funcionalidad de "Agregar un nuevo producto" en la aplicación web del ejercicio 1 utilizando el lenguaje de programación de tu preferencia.



- **Ejercicio 8:** Implementa la lógica de negocio para la funcionalidad de "Agregar un nuevo producto" en la aplicación web del ejercicio 1.



5. Pruebas:

- **Ejercicio 9:** Define un conjunto de pruebas unitarias para la funcionalidad de "Agregar un nuevo producto" en la aplicación web del ejercicio 1.

Caso de Prueba:

Nombre: Prueba de Guardado de Producto

Descripción: Verificar que el sistema guarde correctamente un nuevo producto.

Precondiciones: El sistema está en la pantalla de agregar producto con el formulario visible.

Pasos:

- Simular el llenado del formulario con datos válidos.
- Ejecutar la función de guardar producto.
- Verificar que el producto se ha guardado correctamente en la base de datos.

Resultado Esperado: El producto se guarda correctamente y se muestra un mensaje de confirmación.

- **Ejercicio 10:** Ejecuta pruebas de integración para la funcionalidad de "Agregar un nuevo producto" en la aplicación web del ejercicio 1.

Caso de Prueba:

Nombre: Integración de Guardado de Producto

Descripción: Verificar que el módulo de guardado interactúe correctamente con el sistema de base de datos y la interfaz de usuario.

Precondiciones: El sistema está en la pantalla de agregar producto con el formulario visible.

Pasos:

Simular el llenado del formulario con datos válidos.

Ejecutar la función de guardar producto.

Verificar que los datos se han almacenado correctamente en la base de datos.

Confirmar que se muestra un mensaje de éxito al usuario.

Resultado Esperado: La integración entre la interfaz de usuario, el módulo de guardado y la base de datos es correcta y el producto se guarda exitosamente.

6. Despliegue del Programa:

- **Ejercicio 11:** Definir un plan de despliegue para la aplicación web del ejercicio 1.
 - Utilizar un entorno de desarrollo local para realizar pruebas unitarias y de integración.
 - Asegúrate de que todas las pruebas unitarias e integración hayan sido ejecutadas satisfactoriamente antes de proceder al despliegue.
 - Configurar un ambiente de pre-producción similar al ambiente de producción.
 - Realizar pruebas finales de la aplicación en este ambiente para asegurarte de que todo funciona correctamente antes de ir a producción.
 - Elegir un proveedor de servicios en la nube o un servidor físico para hospedar la aplicación web.
 - Configurar el entorno de ejecución adecuado (por ejemplo, servidor web, base de datos, entorno de ejecución para la aplicación).

Proceso de Despliegue:

- Crear un proceso de despliegue automatizado si es posible (por ejemplo, usando herramientas como Docker, Jenkins, Gitlab CI/CD, etc.).
- Asegurarse de tener una estrategia de rollback en caso de que algo salga mal durante el despliegue.
- Configurar herramientas de monitoreo para supervisar el rendimiento de la aplicación y la infraestructura.
- Establecer un plan de mantenimiento regular para aplicar actualizaciones de seguridad y mejoras.
- **Ejercicio 12:** Despliega la aplicación web del ejercicio 1 en un servidor de producción.

Preparación:

- Asegurarse de que el código de la aplicación esté correctamente versionado y listo para desplegar (por ejemplo, utilizando Git).
- Acceder al servidor de producción y configurar el entorno según lo planificado (servidor web, base de datos, configuraciones de red, etc.).
- Copiar los archivos de la aplicación al servidor de producción.
- Configurar las variables de entorno necesarias (por ejemplo, configuración de base de datos, variables de seguridad).
- Realizar pruebas post-despliegue para asegurarte de que la aplicación funciona correctamente en el entorno de producción.
- Monitorear la aplicación después del despliegue para identificar cualquier problema o degradación de rendimiento.
- Realizar ajustes según sea necesario y documentar cualquier cambio realizado.
- Comunicar el despliegue a los interesados y actualizar la documentación técnica y de usuario según sea necesario.

7. Mantenimiento:

- **Ejercicio 13:** Definir un plan de mantenimiento para la aplicación web del ejercicio 1.
 - Implementar herramientas de monitoreo para supervisar el rendimiento de la aplicación, la disponibilidad del servidor y los errores del sistema.
 - Establecer alertas para responder rápidamente a cualquier problema que surja.
 - Programar actualizaciones regulares para la aplicación y sus componentes (frameworks, bibliotecas, sistemas operativos) para mantener la seguridad y la estabilidad.
 - Implementar parches rápidamente para resolver vulnerabilidades críticas.
 - Establecer un plan de respaldo regular para asegurar los datos críticos de la aplicación.
 - Practicar la recuperación de datos para asegurarte de que el proceso sea efectivo en caso de pérdida de datos.
 - Realizar ajustes periódicos para mejorar el rendimiento de la aplicación, como la optimización de consultas de base de datos, la optimización de imágenes y la mejora de la velocidad de carga.
 - Documentar todos los cambios realizados en la aplicación y en la infraestructura.
 - Implementar un proceso de gestión de cambios para evaluar y aprobar modificaciones antes de implementarlas en producción.
- **Ejercicio 14:** Implementa una corrección de errores para un problema detectado en la aplicación web del ejercicio 1.
 - Identificación del Problema:
 -
 - Utilizar herramientas de monitoreo y reportes de usuarios para identificar el problema específico en la aplicación.

- Crear un entorno de pruebas que refleje el entorno de producción para reproducir el error de manera controlada.
- Analizar el código afectado para identificar la causa raíz del error.
- Desarrollar una solución efectiva y realizar pruebas unitarias para asegurar que la corrección no afecte otras partes de la aplicación.
- Implementar la corrección en el entorno de pruebas y realizar pruebas de integración para verificar que el error se haya resuelto correctamente.
- Una vez verificado en el entorno de pruebas, desplegar la corrección en el entorno de producción siguiendo el plan establecido en el Ejercicio 11.
- Realizar pruebas post-implementación para asegurar que la corrección haya sido exitosa y no haya introducido nuevos problemas en la aplicación.

8. Nos preparamos para nuevos retos

- **Ejercicio 15:** Arme un equipo de trabajo y defina los roles para realizar los ejercicios anteriores para un futuro dominio de aplicación relacionado con inteligencia artificial generativa.

Equipo de Trabajo:

- Líder del Proyecto: Responsable de la coordinación general del proyecto y la comunicación con los interesados.
 - Responsabilidades:
 - Coordinación general del proyecto.
 - Comunicación con los interesados y la alta dirección.
 - Gestión de recursos y planificación del proyecto.
- Desarrolladores: Encargados de implementar las soluciones técnicas, incluyendo el desarrollo de modelos de IA generativa y su integración con la aplicación web.
 - Responsabilidades:
 - Implementación de soluciones técnicas basadas en inteligencia artificial generativa.
 - Desarrollo y optimización de algoritmos y modelos de IA.
 - Integración de los modelos de IA con la aplicación web.
- Ingeniero de Datos: Responsable de la gestión y procesamiento de datos necesarios para el entrenamiento y despliegue de los modelos de IA.
 - Responsabilidades:
 - Gestión y procesamiento de datos necesarios para el entrenamiento y evaluación de modelos de IA.
 - Implementación de pipelines de datos y sistemas de almacenamiento eficientes.
 - Colaboración con el especialista en IA para garantizar la calidad de los datos utilizados en los modelos.

- Especialista en IA: Experto en inteligencia artificial generativa, responsable de diseñar y entrenar los modelos de IA necesarios.
 - Responsabilidades:
 - Diseño y desarrollo de modelos de inteligencia artificial generativa.
 - Entrenamiento y evaluación de modelos utilizando técnicas avanzadas de machine learning y deep learning.
 - Optimización y mejora continua de los modelos para maximizar la precisión y eficiencia.
- QA / Tester: Encargado de realizar pruebas exhaustivas
 - Responsabilidades:
 - Planificación y ejecución de pruebas exhaustivas de los modelos de IA implementados.
 - Validación de la calidad, precisión y fiabilidad de los resultados generados por los modelos.
 - Colaboración con el equipo de desarrollo para identificar y corregir posibles problemas.
- Operaciones / DevOps: Responsable de la configuración y mantenimiento de la infraestructura necesaria para el despliegue de los modelos de IA.
 - Responsabilidades:
 - Configuración y mantenimiento de la infraestructura necesaria para el despliegue de modelos de IA.
 - Implementación de pipelines de CI/CD para automatizar el despliegue y la actualización de modelos.