

Desarrollar software a partir de la integración de sus módulos componentes

Daniel Felipe Lozano Naranjo

Servicio Nacional de Aprendizaje

Centro Latinoamericano de Especies Menores – Regional Valle

Análisis y Desarrollo de software - Ficha: 2721413

Bogotá Colombia

Octubre del 2024

Desarrollar software a partir de la integración de sus módulos componentes

Requisitos funcionales

1. Crear y eliminar usuarios
2. Agregar y/o eliminar insumos
3. Informes de calidad
4. Área de trabajo

Roles de usuario

1. **Administrador:** Es aquel actor con todos los permisos otorgados sobre el sistema
2. **Calidad:** Es aquel usuario con permiso, otorgado por el administrador del sistema, para crear, modificar y/o eliminar reportes de calidad. Adicionalmente tener acceso al área de calidad con permisos de visualización para identificar los pedidos que están en marcha y demás validadores correspondientes del área designada.

Arquitectura del sistema

- **Front - end:** Para este proyecto se implementan tecnologías tales como HTML5, CSS3, JavaScript y React js.
- **Back - end:** Para la implementación del back-end se está trabajando con Node.js.
- **Base de datos:** Se implementa por ahora MySQL para almacenar toda la información proporcionada por los roles participantes en el sistema.

Elección de herramientas de desarrollo

Para el desarrollo de nuestro software he elegido utilizar como herramienta de desarrollo el aplicativo Visual Studio Code por sus amplios beneficios y compatibilidades con muchos lenguajes de programación y por su constante mejora en la aplicación de gadgets útiles.

Como herramienta de versionamiento elegimos utilizar GIT y como almacenamiento de repositorios GITHUB ya que en la actual industria tecnológica, son las opciones más comunes que suelen encontrarse en el ámbito laboral actual.

Identificación de la metodología de desarrollo

Para este proyecto, se va a implementar la metodología SCRUM por sus altos estándares de calidad, alta flexibilidad y por ser una metodología ágil ampliamente utilizada en la industria tecnológica actual.

Casos de uso

1. Crear y eliminar usuarios

Caso de uso	Crear y eliminar usuarios	
Descripción	El administrador podrá y será el único autorizado por el sistema para crear nuevos usuarios y eliminarlos de ser necesario.	
Precondición	Ya debe existir al menos un usuario Administrador activo	
Secuencia normal	Paso	
	1	El administrador ingresa al sistema para crear el usuario
	2	El sistema carga formulario con datos básicos de registro (Cedula, Nombre completo, número celular y rol asignado)
	3	El administrador ingresa los datos y ejecuta el registro
	4	El sistema guarda la información y asigna el rol al nuevo usuario

	5	El sistema asigna una clave aleatoria y un usuario con el nombre de cada persona.
Post condición	El usuario queda registrado y debe cambiar la clave en su primer inicio de sesión.	
Excepciones	Paso	Acción
	1	El sistema detecta la duplicidad de usuarios e impide el registro
	2.	El sistema impedirá el registro si se supera el límite máximo de 10 usuarios registrados en el sistema excluyendo a los administradores.

Caso de uso 2: Agregar y/o eliminar insumos

Caso de uso	Agregar y/o eliminar insumos	
Descripción	El administrador puede agregar nuevas materias primas a la base de datos o eliminarlas.	
Actores	Administrador	
Precondición	Ya debe existir al menos un insumo registrado en el sistema para eliminarlo y debe tener nombre y cantidad del insumo para agregarlo.	
Secuencia normal	Paso	
	1	El administrador ingresa al sistema para agregar o eliminar insumo

	2	El sistema carga los insumos
	3	El administrador elige el insumo y le da clic en eliminar o en agregar.
	4	El sistema guarda la información y asigna el nuevo estado del insumo.
Post condición	El sistema se actualiza inmediatamente mostrando el estado del producto	
Flujo Alternativo	Paso	Acción
	1	Si el administrador cancela el proceso, se termina el caso de uso

Caso de uso 3: Informes de calidad

Caso de uso	Informes de calidad	
Descripción	El administrador puede consultar, archivar, almacenar y/o modificar informes del área de calidad.	
Actores	Administrador	
Precondición	Ya debe existir un producto en proceso de fabricación para poder ejecutar las acciones de calidad.	
Secuencia normal	Paso	
	1	El administrador ingresa al sistema para consultar los informes de calidad

	2	El sistema carga en orden cronológico los informes de calidad
	3	El administrador le da clic en las opciones del informe y elige la opción requerida (Consultar, archivar, almacenar, editar)
	4	El sistema procesa la orden y muestra el resultado
Post condición	El sistema guarda un histórico de movimientos realizados por el administrador	
Flujo Alternativo	Paso	Acción
	1	El sistema permite devolverse a la acción anterior en caso de retracto

Caso de uso 4: Área de trabajo

Caso de uso	Área de trabajo
Descripción	El sistema tendrá un área de trabajo, donde se identificará en orden cronológico y por orden de importancia, los trabajos que se están realizando en el momento de la consulta.
Actores	Administrador
Precondición	Ya debe existir un producto en producción para poder visualizar el área de trabajo.

Secuencia normal	Paso	
	1	El administrador ingresa al sistema para consultar el área de trabajo
	2	El sistema carga el área de trabajo y muestra los productos en fabricación
	3	El administrador le da clic en el producto y revisa el estado del mismo
	4	El sistema muestra la fecha de creación y el estado
Post condición	Una vez finalizada la fabricación del producto, el sistema habilita la opción de finalizar el pedido.	
Flujo Alternativo	Paso	Acción
	1	El sistema permite eliminar pedidos en caso de que por órdenes directas del cliente o del área administrativa, no se continúe con la fabricación del pedido.

Diagrama de Clases

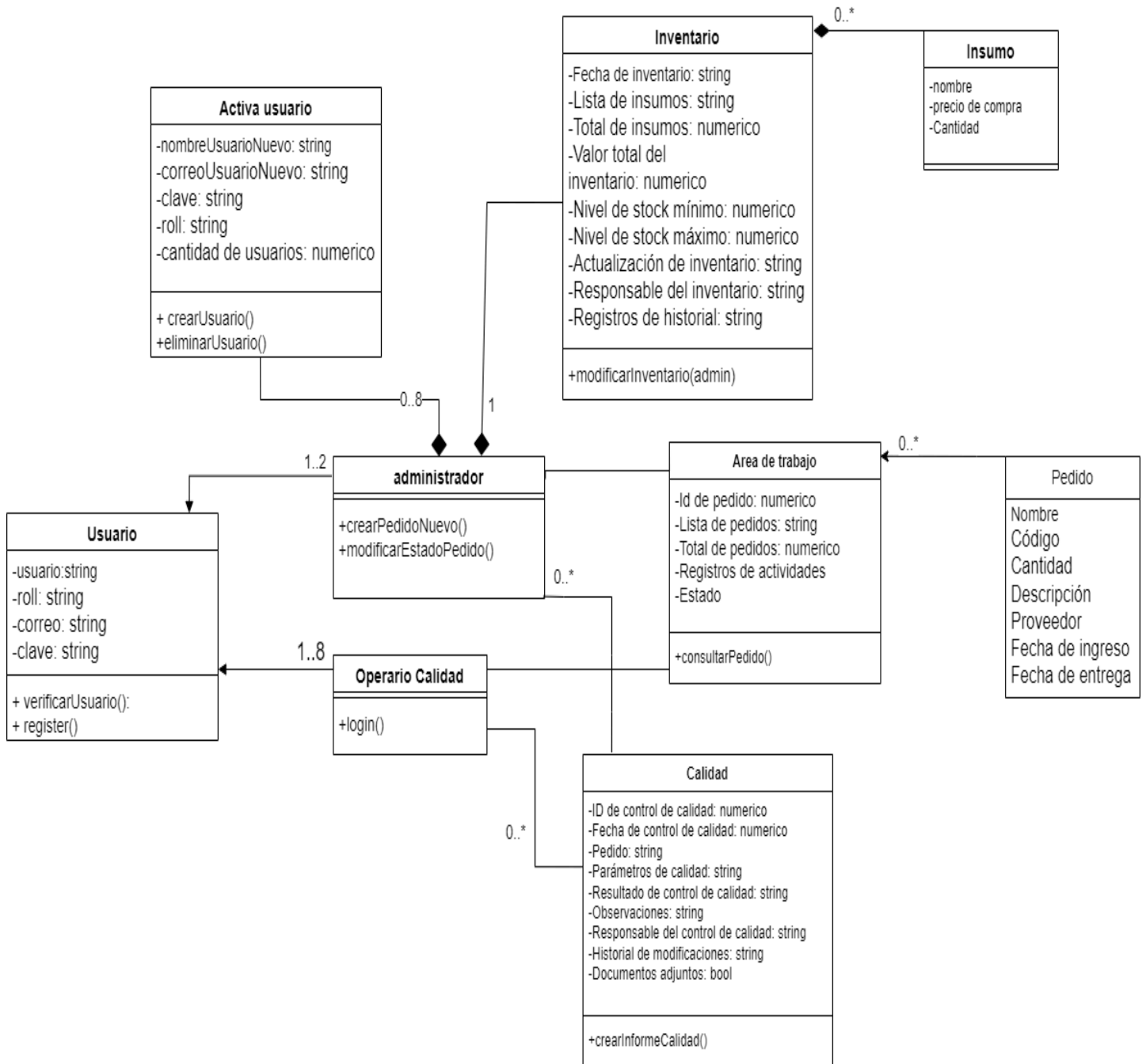
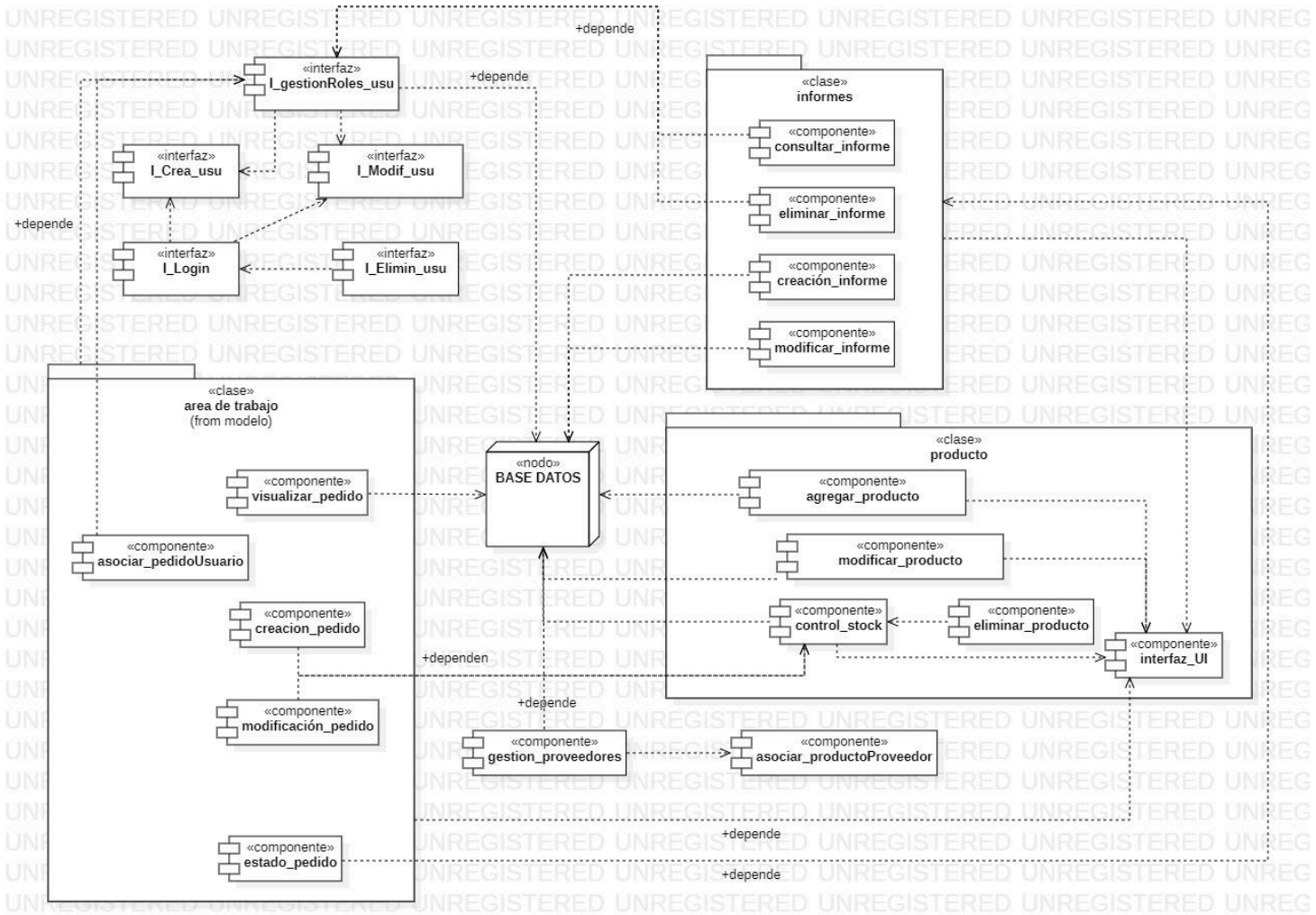
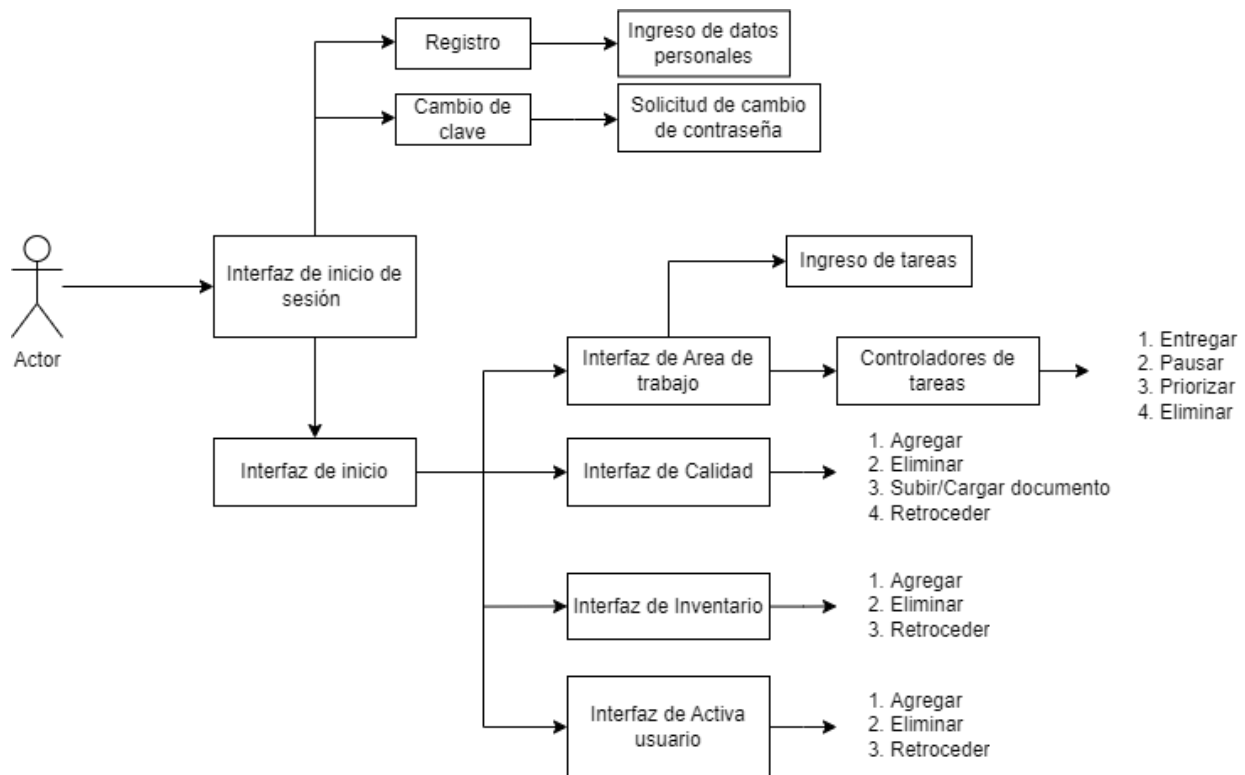


Diagrama de componentes



Mapa de navegación del sistema informático A&N Associates.



Mecanismos de seguridad para mi proyecto web

1. Autenticación basada en Tokens (JWT)

Estado: No implementado aún. En el sistema de login he trabajado con autenticación básica sin tokens. Implementar JWT es una mejora importante para manejar sesiones seguras y permitir la expiración de sesión en el backend y frontend.

2. Validación de Datos Entrantes y Sanitización

Estado: En proceso. He validado que los datos se envíen correctamente desde el frontend y sean procesados por la API. Sin embargo, implementar una

validación más robusta en el backend sería ideal, utilizando librerías como express-validator o Joi.

3. Rate Limiting y Protección contra Fuerza Bruta

Estado: No implementado. Esta es una mejora que podría considerar para limitar intentos de login, evitando posibles ataques de fuerza bruta en la aplicación.

4. Cifrado de Contraseñas

Estado: No implementado. Actualmente, almaceno las contraseñas en texto plano en la base de datos. Esto es crucial de actualizar.

5. Consultas Preparadas para Prevenir SQL Injection

Estado: Implementado en cierta medida. Al usar consultas en Node.js con parámetros (?), estoy evitando la inyección SQL. Sin embargo, verificar que todas las consultas sean seguras es clave.

6. Variables de Entorno

Estado: En uso parcial. Estoy usando variables de entorno para configurar la base de datos, lo cual es correcto. Es importante seguir configurando correctamente todos los datos sensibles para evitar que estos aparezcan en el código público.

Voy a detallar cómo he estructurado mi proyecto hasta ahora en cuanto a lenguajes de programación, librerías, frameworks, configuraciones de servidor y documentación de ambientes.

1. Lenguajes de programación utilizados

En mi proyecto, he trabajado principalmente con:

- JavaScript: Tanto en el frontend como en el backend, utilizando React para la interfaz de usuario y Node.js para el servidor.
- SQL: Para manejar la base de datos MySQL, donde guardo los datos de usuario y gestiono el acceso a la información.

2. Librerías necesarias en cada capa de la aplicación y cuáles ya he usado

Para mi proyecto, he utilizado algunas librerías clave, y hay otras que podría agregar para robustecer la seguridad y funcionalidad.

Frontend:

- Axios: Para manejar las peticiones HTTP entre el frontend y el backend. Ya lo he implementado para la autenticación y el manejo de datos.
- React Router: Aún no he implementado esta librería, pero sería útil para manejar las rutas de manera ordenada en una aplicación de varias páginas.
- PropTypes: Para realizar la validación de tipos de datos de los props; aunque no lo he implementado, ayudaría a manejar errores de datos en componentes.

Backend:

- Express: He usado Express como el framework principal del backend.
- bcrypt: Aunque aún no he implementado el cifrado de contraseñas, bcrypt sería una buena opción para proteger los datos sensibles.
- jsonwebtoken (JWT): Para implementar autenticación segura en las sesiones, algo que también debo integrar.
- dotenv: Esta librería ya la he usado para manejar variables de entorno y asegurar que datos sensibles, como las credenciales de la base de datos, no estén expuestos.

Base de Datos:

No necesito muchas librerías adicionales en la base de datos, pero asegurarme de que las consultas estén bien estructuradas y usar SQL parametrizado (ya implementado) es fundamental para la seguridad.

3. Frameworks en cada capa de la aplicación y cuáles ya he usado

Frontend:

- React: Es el framework principal que uso para la creación de interfaces de usuario. Ya lo he implementado y lo estoy combinando con hooks para manejar el estado y los efectos.

Backend:

- Express: En el backend, he usado Express junto con Node.js para definir mis rutas, manejar la lógica de la API y conectar con la base de datos.

Base de Datos:

- MySQL: Para gestionar los datos de usuario; es el sistema de gestión de bases de datos que he elegido y ya configurado.

4. Configuraciones de servidores y de bases de datos**Servidor de Backend (Node.js con Express):**

- Puerto: Configuro el puerto en el archivo principal para que el servidor se ejecute en el puerto 3000 (o en el puerto definido en `process.env.PORT`).
- dotenv: Uso esta librería para gestionar las credenciales de la base de datos, asegurándome de que estén protegidas en un archivo `.env` y no expuestas en el código.

Base de Datos (MySQL):

Configuración de conexión: La conexión se ha configurado con los datos de usuario, contraseña y base de datos en el archivo `.env`.

Estructura de la base de datos: He estructurado la tabla de usuarios en `loginapi`, con columnas como `id`, `usuario`, y `clave`.

Bugs del sistema en desarrollo

- Actualmente presento el problema de configuración para los botones del componente Botones.jsx ya que los botones Pausar, Priorizar y Entregar no están funcionando de forma correcta. Al momento de ejecutarlos elimina los valores del producto seleccionado y debería solo cambiar el estado del mismo.
- Otro error al cual me veo enfrentado y no encuentro la respuesta es que se pierde la conexión a las URL's de mi API cuando el usuario recarga la pagina en cualquier momento, haciendo que se pierda la conexión y se genere un error y no permita retomar la página.

Links del proyecto: <https://github.com/dano9311/projectGA81096-AA1-EV01>