

# Software de gestión de producción con enfoque en la calidad del producto

Fabian Galindo, Sebastián Hernández, David Viracachá  
No. de Equipo Trabajo: 9

## I. INTRODUCCIÓN

La calidad es un parámetro fundamental que debe estar presente en cualquier bien o producto de consumo. Esta se define por la norma ISO 9000 como el grado en el que las características específicas de un objeto cumplen con un conjunto de requisitos. Teniendo en cuenta que la calidad puede estar implícita en muchos aspectos como los proveedores de las materias primas, los procesos de producción y su distribución, surge la necesidad de encontrar formas de garantizarla. Para este propósito, se han adquirido distintas metodologías a nivel mundial las cuales requieren de, entre otras acciones, de obtener, monitorear y procesar información en distintas etapas clave del proceso de producción; por esta razón la informática puede ser especialmente útil en términos de la eficiencia y confiabilidad con la que se maneja este conjunto de datos, así como en su presentación a un empresario que busca asegurar la calidad de su producto.

## II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Las empresas productoras deben asegurar la calidad de sus productos para cumplir con estándares gubernamentales e internacionales que garantizan la seguridad del consumidor, sin mencionar que, también, la calidad mantiene a un producto competitivo en el mercado. Este ejercicio requiere de un manejo adecuado y preciso de información relacionada al proceso de producción, por esta razón, se busca implementar un software cuyo propósito general es monitorear este conjunto de procesos y factores, brindando así una visión general de la producción que facilite la toma de decisiones destinadas a lograr la calidad del producto, además de permitir la exportación de la información obtenida para simplificar el proceso de certificación de calidad ante los entes de control respectivos.

## III. USUARIOS DEL PRODUCTO DE SOFTWARE

Habrán tres tipos de usuario.

### Administrador y operario

Tiene acceso completo a las funcionalidades, es decir, puede crear y gestionar producciones, agregar materias primas y crear registros.

Puede acceder a la información del proceso y avanzar en sus etapas.

### Estándar

Puede ver el estado de las producciones y generar registros de ser necesario.

## IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

### Crear producción:

Se realiza la creación de una producción, con sus respectivas etapas, parámetros de calidad y materias primas. Un usuario accede a la funcionalidad a partir de un menú, se le facilita al usuario una plantilla para que ingrese los datos de manera correcta. La producción se guardará en una lista de producciones.

Requerimientos funcionales:

Permitirle al usuario ingresar información para llevar a cabo la creación una producción, sus etapas de producción, parámetros y materias primas.

### Crear registro:

Se realiza un resumen de la producción que se haya escogido. El usuario puede ver todos los detalles de la respectiva producción. Este registro no se puede editar.

Requerimientos funcionales:

Se obtienen los datos de la clase producción y se debe exportar a un formato legible para el usuario.

### Gestionar producción:

Se le brinda al usuario una interfaz en la que se muestra una visión general de la producción y sus etapas y en donde podrá seleccionar entre diferentes acciones para realizar sobre su producción como abrir y cerrar etapas, generar reportes parciales o añadir comentarios.

Requerimientos funcionales.

Se debe seleccionar una producción, imprimir su estado general y dar la capacidad a un usuario de ejercer acciones sobre la producción.

### Validar parámetros de calidad:

El valor ingresado debe ser comparado con los rangos establecidos previamente, si el valor está en el rango cumple con el parámetro, de lo contrario no cumple, y se le informa al usuario.

Requerimiento funcional:

El usuario ingresa un valor, se le notifica si cumple con el parámetro. De lo contrario, no podrá avanzar, se obliga al usuario a revisar el proceso de producción y a corregirlo.

## Crear Usuario:

Se realiza la creación del usuario. Una persona que no posea un usuario tiene la opción de la creación de un usuario. Se le pide al usuario ingresar algunos datos y una contraseña. Se realiza esta acción una única vez. Si el usuario ya existe no se permite la acción. Se pueden crear usuarios que tengan diferentes permisos para realizar acciones dentro del programa

Requerimientos funcionales:

Se debe crear un usuario y agregarlo en la base de datos de usuarios. Se debe comprobar que el usuario creado no exista previamente. Se debe asignar un rol al usuario.

## V.DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Se realiza el mockup del proyecto utilizando la herramienta Balsamiq. En las imágenes se muestra algunas funcionalidades del Software de la siguiente manera:

- Inicio de sesión (figura 1)
- Panel de control (figura 2)
- Crear producción (figura 3)
- Buscador de producción (figura 4)
- Gestión de producciones (figura 5)
- Gestión de parámetros (figura 6)
- Creación de registro de producción (figura 7)
- Perfil de usuario (figura 8)



Figura 1. Iniciar sesión.



Figura 2. Panel de control

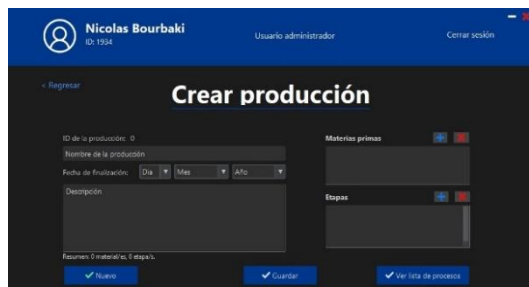


Figura 3. Crear producción.



Figura 4. Buscador de producción



Figura 5. Gestión de producción.

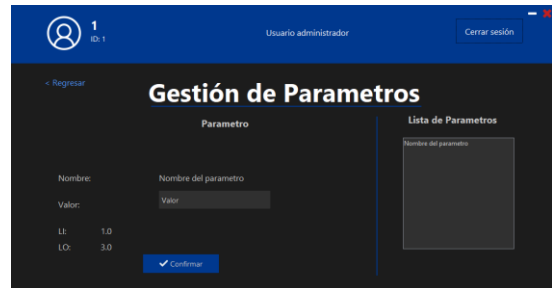


Figura 6. Gestión de parámetros.



Figura 7. Creación de registro.

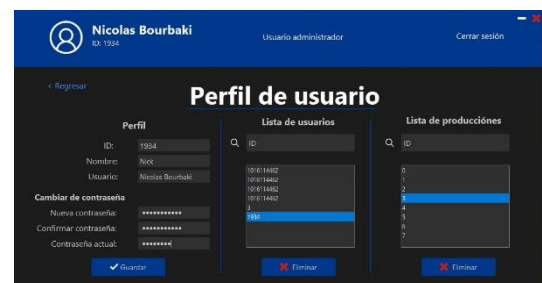


Figura 8. Perfil de usuario.

## VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

El software se desarrollada en NetBeans 11.0 o mayor e IntelliJ IDEA en el lenguaje de programación Java 14.0.1(Java HotSpot(TM) 64-Bit Server VM 14.0.1+7). También se usa la plataforma GitKraken para comparar los aportes de cada integrante del equipo de trabajo y decidir qué elementos pasaran a ser parte de la versión del prototipo. Adicional, esta plataforma permite almacenar el proyecto en un repositorio de GitHub con el fin de que cada integrante tenga el código software actualizado.

A continuación, se dan a conocer los requisitos mínimos del sistema para el software:

**Sistema Operativo:** Windows 10 para 64 bits.

**Procesador:** Procesador a 3.0 GHz o más rápido.

**RAM:** 4GB.

**Espacio en disco duro:** 100 MB.

**Tarjeta gráfica:** DirectX 9 o posterior con controlador WDDM 1.0

**Pantalla:**1024x600

## VII. PROTOTIPO DE SOFTWARE INICIAL

Para esta entrega se manejó el siguiente repositorio de GitHub: [https://github.com/fsgalindope/Proyecto\\_estructuras\\_de\\_datos\\_2020-1\\_grupo\\_9/tree/Segunda\\_Entrega](https://github.com/fsgalindope/Proyecto_estructuras_de_datos_2020-1_grupo_9/tree/Segunda_Entrega)

## VIII. PRUEBAS DEL PROTOTIPO Y ANALISIS

En la primera entrega se planteó usar entre lista enlazadas o listas implementadas por arreglos dinámicos para almacenar usuarios y producciones, lo cual como no se van a realizar eliminaciones se escoge los arreglos dinámicos. Para esta entrega se implementó una nueva estructura de datos que es el árbol AVL. El cual se va a probar las siguientes funcionalidades:

- Insertar n elementos al árbol (grafica 1 y tabla 1)
- Eliminar una hoja, Inicio de sesión (grafica 2 y tabla 2)
- Encontrar el mínimo de un árbol de n elementos (grafica 3 y tabla 3)
- Encontrar el máximo de un árbol de n elementos (grafica 1 y tabla 1)
- Encontrar un elemento aleatorio de un árbol de n elementos (grafica 1 y tabla 1)

Insertar en un Avl	
Datos	Tiempo(ms)
10000	542
100000	107142
250000	683697
500000	2920283
1000000	11879777
10000000	No aplica

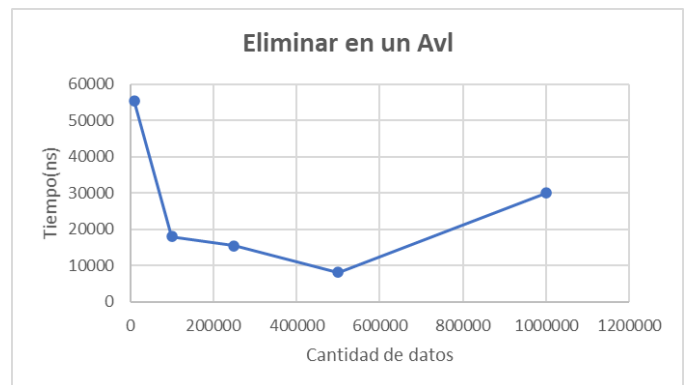
Tabla 1.Insertar en un AVL Datos vs Tiempo(ms)



Grafica 1. Insertar en un AVL Datos vs Tiempo(ms)

Eliminar en un Avl	
Datos	Tiempo(ns)
10000	55400
100000	18000
250000	15500
500000	8100
1000000	30000
10000000	No aplica

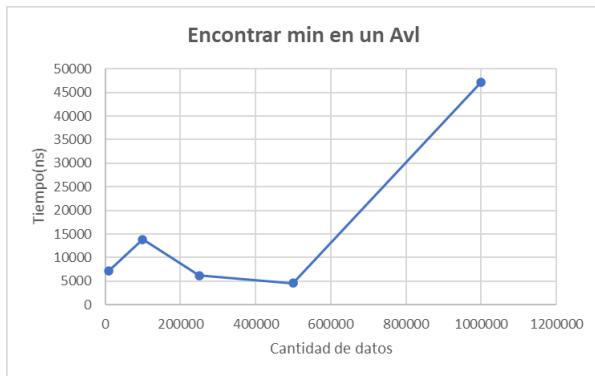
Tabla 2. Eliminar un elemento en un AVL Datos vs Tiempo(ns).



Grafica 2 Eliminar un elemento en un AVL Datos vs Tiempo(ns).

Encontrar min en un Avl	
Datos	Tiempo(ns)
10000	7200
100000	13900
250000	6200
500000	4600
1000000	47200
10000000	No aplica

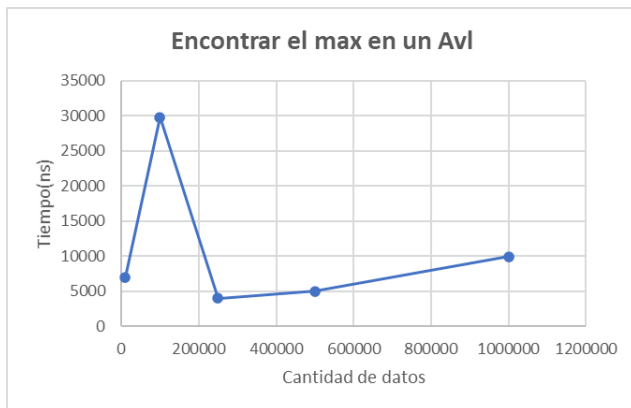
Tabla 3 Encontrar el min elemento del AVL Datos vs Tiempo(ns).



Gráfica 3 Encontrar el min elemento del AVL Datos vs Tiempo(ns).

Encontrar el max en un AVL	
Datos	Tiempo(ns)
10000	7000
100000	29800
250000	4000
500000	5000
1000000	9900
10000000	No aplica

Tabla 4 Encontrar el max elemento del AVL Datos vs Tiempo(ns).



Gráfica 4 Encontrar el max elemento del AVL Datos vs Tiempo(ns).

Encontrar aleatorio en un AVL	
Datos	Tiempo(ms)
10000	45300
100000	17500
250000	9900
500000	2400
1000000	6500
10000000	No aplica

Tabla 5 Encontrar un elemento aleatorio del AVL Datos vs Tiempo(ns).



Gráfica 5 Encontrar un elemento aleatorio del AVL Datos vs Tiempo(ns).

No se realizó pruebas para un árbol de búsqueda binario porque el balance del árbol AVL permite asegurar que tendrá un mejor comportamiento es gasto computacional. El árbol AVL de producciones se utiliza en el programa para hacer una búsqueda a partir del nombre de producciones para que sea eficiente al generar registros.

## IX. ANÁLISIS DE COMPARATIVO

En la gráfica 1. Insertar en un AVL Datos vs Tiempo(ms) Podemos observa que se comporta de manera casi cuadrática. Pero haciendo un análisis más preciso podemos concluir que el gasto computacional es  $O(n \cdot \log_2 n)$ .  $\log_2 n$  por cada inserción y  $n$  que es la cantidad de datos que se ingresa. Es la funcionalidad que mas gasta computacionalmente. Pero para este caso no se tiene problema ya que nunca se hacer inserción de una cantidad  $n$  de elementos en un mismo momento. Las gráficas 3, 4, y 5 de encontrar están a escalas de nanosegundo inclusive para un árbol con 1 millón de datos, esto es bueno en nuestra implementación que se basa en la búsqueda de elementos. Las graficas presentan un comportamiento algo diferente a lo esperado que es un  $O(\log_2 n)$  pero si aumentamos la escala no se observa un crecimiento mayor al esperado.

## X. ROLES Y ACTIVIDADES

Sebastian Hernández

- Roles: Coordinador, Técnico
- Actividades realizadas:
  - Programación de la interfaz gráfica: Crear usuarios, crear y buscar producción y generar registro.
  - Desarrollo, manejo y definición de la base de datos.

David Viracachá

- Roles: Investigador, Líder
- Actividades realizadas:
  - Implementación de los árboles y árboles AVL estructuras de datos.
  - Aportaciones en la interfaz grafica

- Aportes en la creación del mockup

Fabian Galindo

- Roles: Observador, Animador
- Actividades realizadas:
  - Programación de la interfaz gráfica: Gestor de producción
  - Edición y revisión del documento
  - Realización de las pruebas
  - Aporte en el desarrollo de las colas

## XI. DIFICULTADES Y LECCIONES APRENDIDAS

Durante el desarrollo del proyecto se encontraron inconvenientes con los computadores ya que generaba fallas al manejar más 1 millón de datos debido su capacidad de procesamiento.

Se aprendió a manejar herramientas que facilitan el desarrollo de software tales como GitKraken adicional se adquiere conocimiento de diferentes funciones para interfaz gráfica y mejoras visuales, entre otros.

Comprendimos la importancia de definir asertivamente la estructura de datos con el fin optimizar los procesos que se llevan a cabo en el software.

Entendimos que las definiciones recursivas de los árboles, aunque no siempre resultan sencillas de conceptualizar, resultan muy útiles en su implementación computacional. También, entendemos que existen estructuras de datos que, aunque no sean tan sencillas y/o triviales y requieran de un mantenimiento más intenso, resultan especialmente útiles en términos de efectividad al momento de realizar ciertas operaciones. Con esto también podemos decir que entre estructuras hay ciertos intercambios de memoria y rendimiento los cuales deben ser tenidos en cuenta al momento de usar cualquiera de ellas.