

SEGUNDA ENTREGA PROYECTO

Second delivery of the project
Segunda entrega del proyecto

García Figueroa Daniel Santiago
Lobatón Galindo Sirley Xiomara
Palacio Berrio Santiago
Politécnico Gran Colombiano
Colombia

Recepción: XX.XX.XXXX
Aceptación: XX.XX.XXXX

DOI: XXXXXX

Resumen

Este apartado debe estar diseñado para invitar al lector a continuar leyendo el documento de tal manera que se debe indicar: la motivación y propósito del trabajo, el método para llevar a cabo lo que va del documento y una idea de los resultados logrados. La extensión máxima es de 300 palabras, y puede dividirse de acuerdo con los siguientes apartados: Objetivo, Métodos y materiales, Resultados y Conclusiones (generalidades)

Palabras clave

En este apartado se deben enunciar de **3 a 6 palabras o descriptores** asociados al contenido del manuscrito. Idealmente en este apartado se deben incluir términos universalmente categorizados y organizados en el área de conocimiento en el cual se publica; estos términos o tesauros -como son conocidos- facilitan la indización del artículo en bases de datos y buscadores especializados por ello se recomienda que no sea una transcripción separada por comas del título del documento. Todas **deben ser escritas en minúscula y separadas por coma (,).**

“Este documento es propiedad intelectual del POLITECNICO GRANCOLOMBIANO, se prohíbe su reproducción total o parcial sin la autorización escrita de la Rectoría. TODO DOCUMENTO IMPRESO O DESCARGADO DEL SISTEMA, ES CONSIDERADO COPIA NO CONTROLADA”.

INTRODUCCIÓN

REPRESENTACIONES GRÁFICAS.

Ilustración 1

Los pilares de la programación orientada a objetos



Nota. La ilustración 1 contiene la representación gráfica los pilares de la programación orientada a objetos, el cual es uno de los conceptos a usar el desarrollo de la aplicación.

Ilustración 2

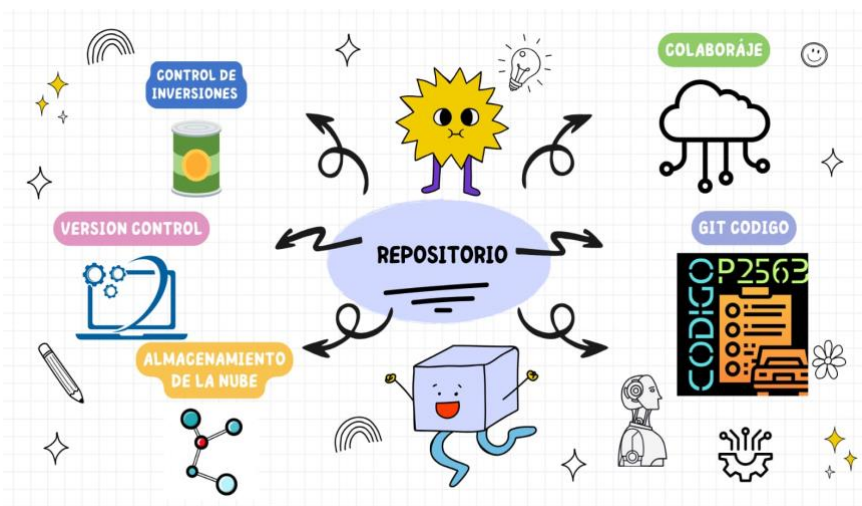
Arquitectura Modelo-Vista-Controlador



Nota. La ilustración 2 contiene la representación gráfica de la arquitectura Modelo-Vista-Controlador, el cual es uno de los conceptos a usar el desarrollo de la aplicación.

“Este documento es propiedad intelectual del POLITECNICO GRANCOLOMBIANO, se prohíbe su reproducción total o parcial sin la autorización escrita de la Rectoría. TODO DOCUMENTO IMPRESO O DESCARGADO DEL SISTEMA, ES CONSIDERADO COPIA NO CONTROLADA”.

Ilustración 3
Repositorios



Nota. La ilustración 3 contiene la representación gráfica de los repositorios, el cual es uno de los conceptos a usar el desarrollo de la aplicación.

Ilustración 4
Java Collection Framework



Nota. La ilustración 4 contiene la representación gráfica de Java Collection Framework, el cual es uno de los conceptos a usar el desarrollo de la aplicación.

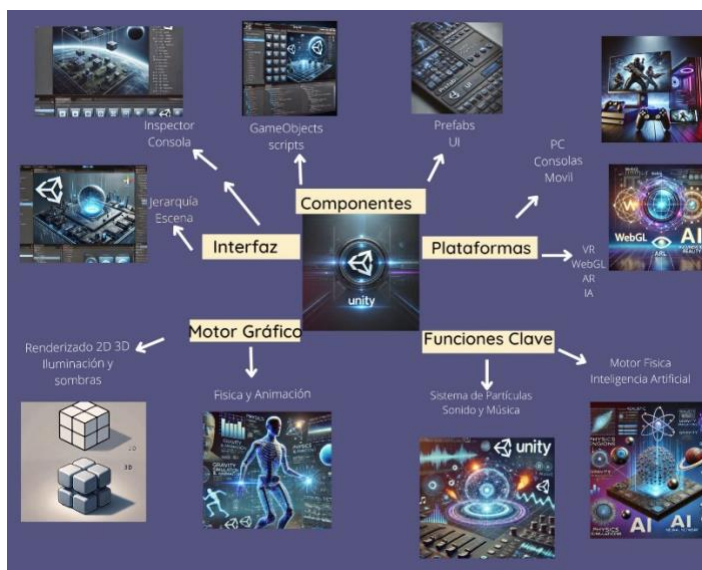
“Este documento es propiedad intelectual del POLITECNICO GRANCOLOMBIANO, se prohíbe su reproducción total o parcial sin la autorización escrita de la Rectoría. TODO DOCUMENTO IMPRESO O DESCARGADO DEL SISTEMA, ES CONSIDERADO COPIA NO CONTROLADA”.

Ilustración 5
Javadoc



Nota. La ilustración 5 contiene la representación gráfica de Javadoc, el cual es uno de los conceptos a usar el desarrollo de la aplicación.

Ilustración 6
Junit



Nota. La ilustración 6 contiene la representación gráfica de Junit, el cual es uno de los conceptos a usar el desarrollo de la aplicación.

“Este documento es propiedad intelectual del POLITECNICO GRANCOLOMBIANO, se prohíbe su reproducción total o parcial sin la autorización escrita de la Rectoría. TODO DOCUMENTO IMPRESO O DESCARGADO DEL SISTEMA, ES CONSIDERADO COPIA NO CONTROLADA”.

ENUNCIADO DEL PROBLEMA.

En una cooperativa agrícola, los productores deben llevar un control preciso de los fertilizantes que utilizan en sus cultivos. Actualmente, la gestión de estos productos se realiza manualmente en hojas de cálculo o registros físicos, lo que dificulta la consulta, actualización y eliminación de datos, generando problemas en la planificación y el control de inventario.

Para solucionar este problema, se desarrollará una aplicación de escritorio en JavaFX basada en el paradigma orientado a objetos y siguiendo la arquitectura Modelo-Vista-Controlador (MVC). La aplicación permitirá a los agricultores registrar, consultar, actualizar y eliminar información sobre los fertilizantes utilizados en sus cultivos. Cada fertilizante contará con atributos como nombre, tipo, fecha de adquisición, cantidad disponible y proveedor.

Este sistema facilitará la gestión eficiente de los fertilizantes, mejorando el acceso a la información y optimizando los recursos agrícolas.

DISEÑO PROTOTIPO VISTA DE USUARIO.

Ilustración 7

Primera versión del prototipo de vista de usuario para la aplicación.



El prototipo muestra una interfaz de usuario titulada "Gestión de fertilizantes". A la izquierda hay un formulario con los siguientes campos: "Nombre del fertilizante" (campo de texto), "Tipo de fertilizante" (menú desplegable), "Proveedor" (campo de texto), "Fecha de adquisición" (campo de texto con icono de calendario), y "Cantidad disponible" (campo de texto). Debajo de estos campos hay un botón "Buscar" y un campo de texto "Buscar fertilizante". En la parte inferior del formulario hay cuatro botones: "Agregar fertilizante", "Actualizar fertilizante", "Eliminar fertilizante" y "Limpiar formulario". A la derecha del formulario hay una tabla con los encabezados "Nombre", "Tipo", "Proveedor" y "Fe". El cuerpo de la tabla muestra el mensaje "Tabla sin contenido".

Nota. La ilustración 1 contiene el diseño de la primera versión de la vista de usuario para la aplicación hecha en Scene Builder.

"Este documento es propiedad intelectual del POLITECNICO GRANCOLOMBIANO, se prohíbe su reproducción total o parcial sin la autorización escrita de la Rectoría. TODO DOCUMENTO IMPRESO O DESCARGADO DEL SISTEMA, ES CONSIDERADO COPIA NO CONTROLADA".

PROPUESTA DE CROGRAMA DE DESARROLLO.

A continuación, se presentará una propuesta de cronograma y actividades para el desarrollo de la aplicación durante el semestre dividido en tres fases, de acuerdo con las entregas establecidas en el documento “Proyecto – paradigmas.pdf” suministrado por el profesor en la plataforma institucional.

Tabla 1
Fase 1. Diseño y planificación (Semanas 1-4)

<i>Semana</i>	<i>Actividad</i>	<i>Descripción</i>
Uno	Diseño de la interfaz en Scene Builder.	Se busca crear el primer prototipo de la vista de usuario.
Dos	Definir requisitos y estructura del CRUD.	Establecer los atributos del fertilizante y sus respectivas operaciones CRUD.
Tres	Planificar la arquitectura a MVC.	Diseñar la estructura del código aplicando MVC.
Cuatro	Primera entrega.	Entregar la primera versión de la vista de usuario para la aplicación.

Nota. La tabla 1 presenta la primera fase del cronograma de desarrollo de la aplicación que tiene como objetivo definir la estructura de la aplicación y establecer la base del desarrollo.

Tabla 2
Fase 2. Modelado y desarrollo inicial (Semana 5-9)

Cinco	Modelado del diagrama de clases UML en Star UML.	Diseñar el diagrama de clases correspondiente a la aplicación con sus respectivas relaciones y métodos.
Seis	Estructuración de las colecciones (operaciones de la funcionalidad CRUD) para la aplicación.	Representar gráficamente las estructuras de datos o colecciones necesarias para el desarrollo de la aplicación.
Siete y ocho	Implementación del modelo	Emigrar el diagrama del proyecto en Star UML a eclipse, y codificar las clases del dominio y la lógica del CRUD.

Nueve	Segunda entrega	Subir al repositorio el diagrama, estructuras bases y avance del programa de la aplicación.
-------	-----------------	---

Nota. La tabla 2 presenta la segunda fase del cronograma de desarrollo de la aplicación que tiene como objetivo crear el modelo UML de la misma y comenzar la implementación de su respectivo código.

Tabla 3
Fase 3. Desarrollo final, pruebas y documentación (Semanas 10-16)

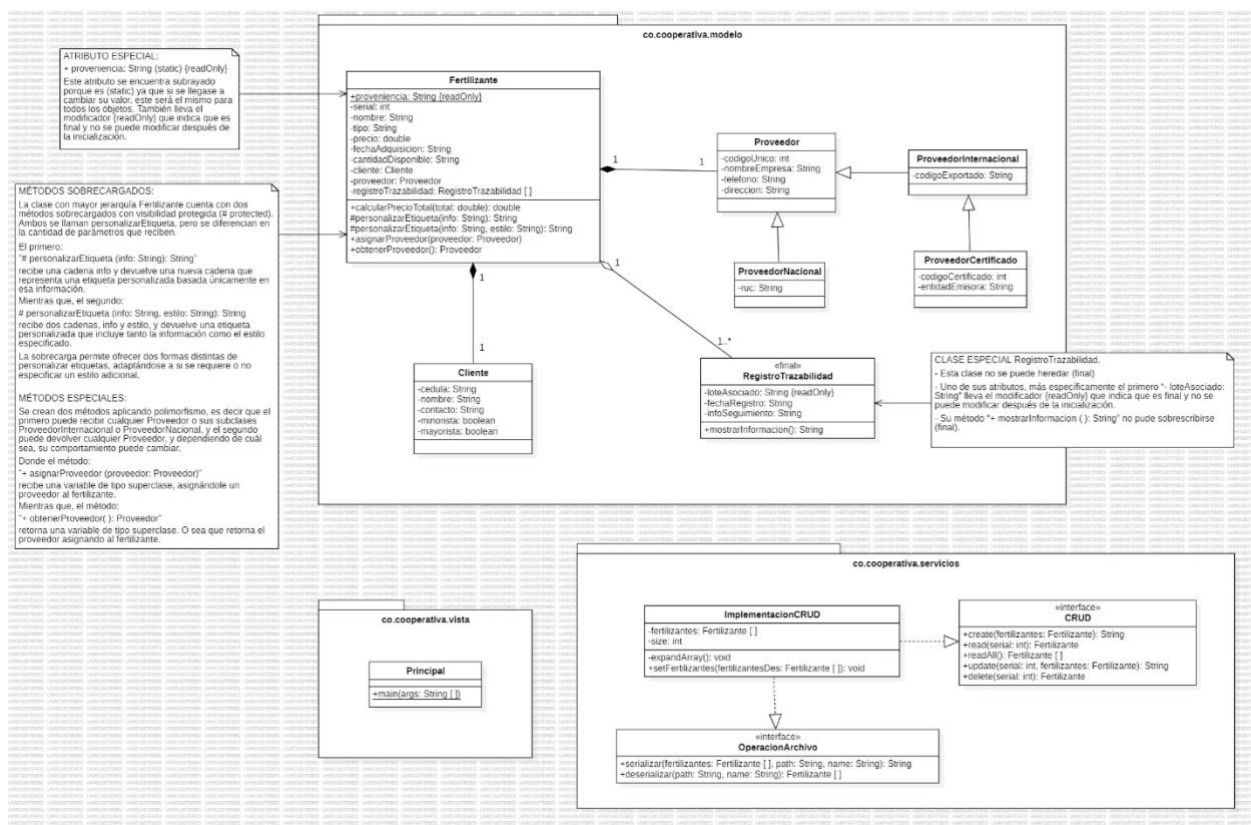
Diez	Implementación de la vista en JavaFx.	Desarrollar la versión final de la interfaz con Scene Builder o código manual
Once	Conectar la interfaz con el controlador.	Implementar eventos y enlazar la UI con la lógica del CRUD.
Doce y trece	Pruebas iniciales, optimización y ajuste.	Validar el funcionamiento de las operaciones CRUD y corregir errores. E igual modo, mejorar la estructura del código y la usabilidad.
Catorce	Elaboración de la documentación.	Redactar el respectivo manual de usuario de la aplicación y documentación técnica.
Quince	Grabación del video	Mostrar la funcionalidad completa de la aplicación (acciones CRUD).
Dieciséis	Tercera entrega	Subir la URL a un repositorio público como Github con el programa final, la documentación, y el video de la aplicación.

Nota. La tabla 3 presenta la tercera fase del cronograma de desarrollo de la aplicación que tiene como objetivo completar el desarrollo de la aplicación, realizar pruebas técnicas y documentar el proceso.

MÉTODO

PRIMER PUNTO:

Ilustración 8
Diagrama de clases UML.

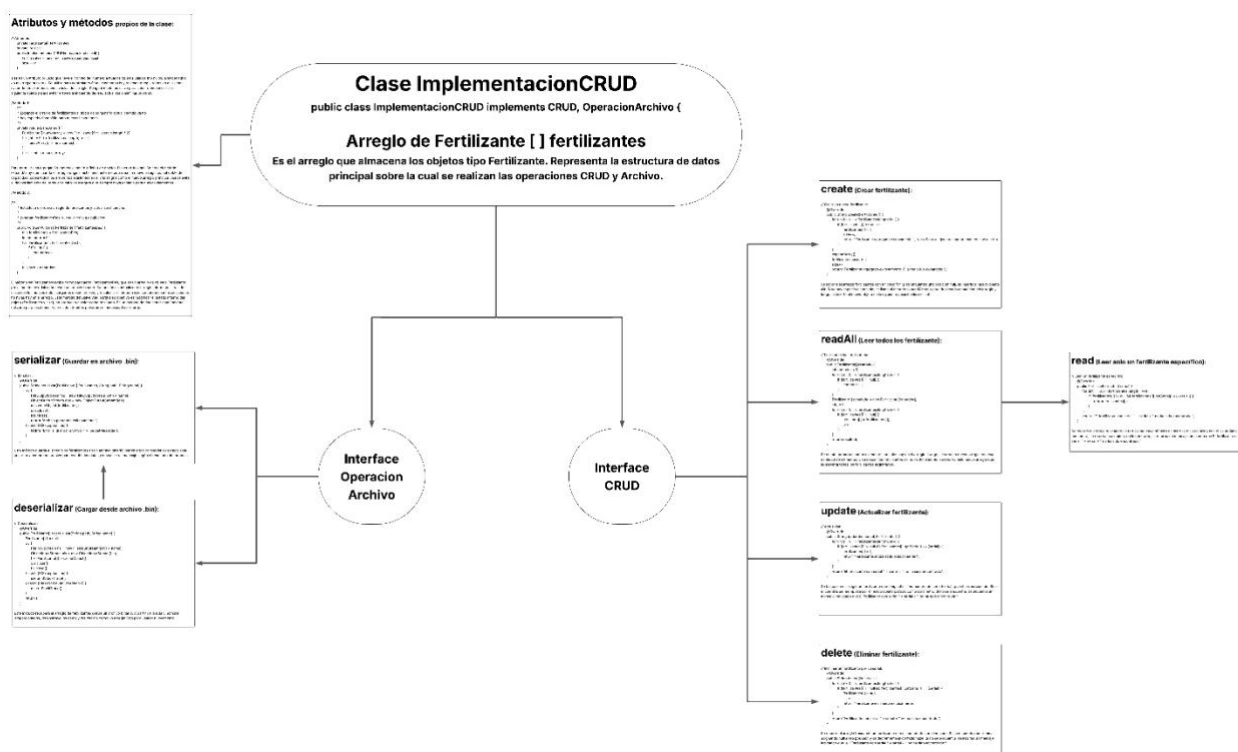


Nota. La ilustración 8 contiene el modelo de la aplicación de gestión de fertilizantes usando el diagrama de clases UML. Para una mejor visualización se recomienda ver desde el repositorio Github del proyecto mediante el enlace que se encuentra en la sección “CORRESPONDENCIA DEL MODELO” de la página 9.

“Este documento es propiedad intelectual del POLITECNICO GRANCOLOMBIANO, se prohíbe su reproducción total o parcial sin la autorización escrita de la Rectoría. TODO DOCUMENTO IMPRESO O DESCARGADO DEL SISTEMA, ES CONSIDERADO COPIA NO CONTROLADA”.

SEGUNDO PUNTO:

Ilustración 9
Gráfico operaciones de funcionalidad CRUD



Nota. La ilustración 9 contiene la representación gráfica de las operaciones sobre la estructura de datos seleccionada. Para una mejor visualización se recomienda ver desde el repositorio Github del proyecto mediante el enlace que se encuentra en la sección “CORRESPONDENCIA DEL MODELO” de la página 9 (A continuación).

TERCER PUNTO:

CORRESPONDENCIA DEL MODELO.

Diseñado con el código del repositorio Github:

<https://github.com/Dany-Garcia/Proyecto-paradigmas-de-programaci-n>

“Este documento es propiedad intelectual del POLITECNICO GRANCOLOMBIANO, se prohíbe su reproducción total o parcial sin la autorización escrita de la Rectoría. TODO DOCUMENTO IMPRESO O DESCARGADO DEL SISTEMA, ES CONSIDERADO COPIA NO CONTROLADA”.

RESULTADOS

Expone los hallazgos y datos resultantes del análisis de datos presentados anteriormente en los métodos, los cuales deben tener coherencia y ética respecto al tratamiento de la pregunta de la investigación planteada.

En este espacio, se realiza la constatación de las hipótesis y resultados de los objetivos de investigación planteados en el estudio argumentando sus diferentes perspectivas/hallazgos.

Se recomienda:

- Evidenciar situaciones o hallazgos con argumentos en hechos facticos.
- No hacer de este espacio o sección una lista de situaciones
- No usar o insertar notas al pie, ni notas al final
- Usar subtítulos que aporten mejoras en la legibilidad del documento

DISCUSIÓN Y CONCLUSIÓN

Interpreta los resultados de la investigación demostrando la relevancia de la investigación e innovación en el área de conocimiento.

Necesariamente, establece recomendaciones a nuevos estudios para su abordaje o continuidad de conformidad a los resultados encontrados.

REFERENCIAS BIBLIOGRÁFICAS

Martin, R. C. (2009). Código limpio: Manual de estilo para el desarrollo ágil de software. Prentice Hall.

Pavón Mestras, J. (2008). El patrón Modelo-Vista-Controlador (MVC). Facultad de Informática, Universidad Complutense de Madrid.

Schildt, H. (2018). Java: The Complete Reference (11ª ed.). McGraw-Hill Education.

Gosling, J., Joy, B., Steele, G., Bracha, G., & Buckley, A. (2014). The Java Language Specification (Java SE 8 Edition). Addison-Wesley.