

SERVICIO NACIONAL DE ADIESTRAMIENTO EN TRABAJO INDUSTRIAL

SENATI



PROYECTO:

DESARROLLO DEL ALGORITMO ID3

DOCENTE:

RUDY FRANKLIN CONDORI QUILLA

PRESENTADO POR:

MARITZA CAMILA PONGO MAMANI

SEMESTRE:

IV

ENLACE DEL REPOSITORIO:

<https://github.com/maritzacamila11pm/Algoritmo-ID3.git>

PUNO – PERU – 2025

INDICE

1. Objetivo General	4
1.1. Objetivos Específicos	
2. Descripción del Sistema	4
2.1. Propósito	
2.2. Funcionalidades Principales	
2.3. Atributos de Entrada	
2.4. Carreras de Salida	
3. Cronograma y Hitos del Proyecto	6
3.1. Calendarización	
3.2. Hitos	
3.2.1. Fase de Planificación y Análisis de Requisitos	
3.2.2. Elaboración de Diagramas y Modelos	
3.2.3. Implementación y Desarrollo del Sistema	
3.2.4. Pruebas y Entrega del Proyecto	
4. Requerimientos	8
4.1. Requerimientos Funcionales	
4.2. Requerimientos No Funcionales	
5. Arquitectura del Sistema	9
5.1. Patrón de Diseño	
5.2. Componentes Principales	
5.2.1. Clase Main	
5.2.2. Clase ArbolID3Visual	

5.2.3. Clase Ejemplo	
5.2.4. Clase NodoID3	
6. Algoritmo ID3 Implementado	10
6.1. Fundamentos Teóricos	
6.2. Proceso de Construcción	
6.3. Características de la Implementación	
7. Diseño del Software	12
7.1. Diagrama de Clase	
7.2. Caso de Uso Principal	
8. Resultados Esperados	13
8.1. Métricas de Calidad	
8.2. Beneficios del Sistema	
9. Conclusiones	14
9.1. Logros Alcanzados	
9.2. Limitaciones Identificadas	
9.3. Trabajos Futuros	
10. Referencias Técnicas	14
10.1. Tecnologías Utilizadas	
10.2. Patrones Implementados	
10.3. Estructuras de Datos	
11. Implementación	15

1. OBJETIVO GENERAL

Desarrollar un sistema inteligente de recomendación de carreras profesionales utilizando el algoritmo de árboles de decisión ID3, que permita a los usuarios obtener sugerencias personalizadas basadas en sus intereses, preferencias laborales, habilidades y nivel de estudios.

1.1.Objetivos Específicos

- Implementar el algoritmo ID3 para construcción automática de árboles de decisión
- Crear una interfaz gráfica intuitiva usando JavaFX para visualización del árbol
- Procesar datos de entrada con atributos relevantes para la elección de carrera
- Generar recomendaciones precisas basadas en patrones de datos históricos
- Visualizar el proceso de decisión de forma clara y comprensible

2. DESCRIPCIÓN DEL SISTEMA

2.1.Propósito

El sistema utiliza inteligencia artificial para analizar las características personales del usuario y recomendar carreras profesionales que mejor se adapten a su perfil. Implementa el algoritmo ID3 (Iterative Dichotomiser 3) para construir un árbol de decisión basado en datos de entrenamiento.

2.2. Funcionalidades Principales

- Visualización de Datos: Muestra los datos de entrenamiento en una tabla interactiva

- Construcción de Árbol: Genera automáticamente un árbol de decisión usando ID3
- Representación Visual: Dibuja el árbol construido en un canvas gráfico
- Análisis de Métricas: Calcula altura, número de nodos y hojas del árbol
- Clasificación: Permite clasificar nuevos casos basados en el árbol construido

2.3. Atributos de Entrada

- Interés Principal: Tecnología, Salud, Arte, Negocios
- Preferencia de Trabajo: Oficina, Laboratorio, Creativo, Campo
- Habilidad Destacada: Cálculo, Precisión, Creatividad, Comunicación
- Nivel de Estudios: Alto, Medio, Básico

2.4. Carreras de Salida

- Ingeniería
- Medicina
- Diseño Gráfico
- Administración
- Psicología
- Biotecnología
- Sistemas
- Agronomía

3. CCRONOGRAMA Y HITOS DEL PROYECTO

3.1. Calendarización

Fecha	Actividad	Responsable	Estado
30 mayo	Análisis y definición de requisitos	Maritza	Completado
31 mayo	Diseño de la arquitectura MVC	Maritza	Completado
1 junio	Implementación algoritmo ID3	Maritza	Completado
2 junio	Desarrollo interfaz JavaFX	Maritza	Completado
3 junio	Pruebas y corrección de errores	Maritza	Completado
4 junio	Documentación final y entrega	Maritza	Completado

3.2. Hitos

DURACIÓN DEL PROYECTO	
INICIO:	30 de mayo de 2025
FINAL:	4 de junio de 2025
DURACIÓN:	6 días

3.2.1. Fase de Planificación y Análisis de Requisitos

- Inicio: 30 de mayo de 2025
- Final: 31 de mayo de 2025
- Duración: 2 días

En esta fase se definieron los objetivos generales y específicos del sistema inteligente de recomendación usando el algoritmo ID3. Se realizó el análisis de los requisitos funcionales y no funcionales, identificando los atributos de entrada, salidas esperadas y las funcionalidades principales. Se elaboró el cronograma general del proyecto y se establecieron los criterios de calidad y alcance.

3.2.2. Elaboración de Diagramas y Modelos

- Inicio: 1 de junio de 2025
- Final: 2 de junio de 2025
- Duración: 2 días

Durante esta fase se desarrollaron los diagramas de clases y casos de uso que describen la arquitectura del sistema basada en el patrón MVC. Se documentaron las responsabilidades de las clases principales y se detalló el flujo del proceso de decisión mediante árboles. Se preparó la documentación visual que respalda el diseño del software.

3.2.3. Implementación y Desarrollo del Sistema

- Inicio: 3 de junio de 2025
- Final: 3 de junio de 2025
- Duración: 1 día

Esta fase se dedicó a la codificación del algoritmo ID3 y la integración con la interfaz gráfica desarrollada en JavaFX. Se implementó la construcción automática del árbol, la visualización en canvas gráfico y la funcionalidad para clasificar nuevos casos. Se aseguraron las condiciones de parada y se probaron las funcionalidades básicas.

3.2.4. Pruebas y Entrega del Proyecto

- Inicio: 4 de junio de 2025

- Final: 4 de junio de 2025
- Duración: 1 día

En esta etapa se realizaron pruebas funcionales para validar la precisión y la generalización del sistema. Se verificó la correcta visualización del árbol, la interacción con los datos y la respuesta del sistema ante nuevos casos. Se preparó la documentación final y se entregó el proyecto completo y funcional.

4. REQUERIMIENTOS

Código	Requerimiento
RF1	El sistema debe construir un árbol de decisión usando el algoritmo ID3 con base en los datos.
RF2	El sistema debe mostrar el árbol de decisión generado en una interfaz gráfica clara y navegable.
RF3	El sistema debe permitir clasificar nuevos casos ingresados y mostrar la recomendación de carrera.
RF4	El sistema debe mostrar los datos de entrenamiento en una tabla interactiva para revisión.
RF5	El sistema debe permitir salir de la interfaz

Código	Requerimiento
RNF1	La interfaz debe ser intuitiva y fácil de usar para usuarios sin conocimientos técnicos.
RNF2	El sistema debe construir y mostrar el árbol en un tiempo razonable (menos de 5 segundos).
RNF3	El sistema debe permitir incorporar más datos sin necesidad de grandes cambios en el código.
RNF4	El software debe ser compatible con Windows, Linux y macOS que soporten Java 8+.
RNF5	El código debe estar bien estructurado para facilitar futuras modificaciones y mantenimiento.

5. ARQUITECTURA DEL SISTEMA

5.1. Patrón de Diseño

El sistema implementa el patrón MVC (Model-View-Controller) adaptado:

- Model: Ejemplo.java, NodoID3.java - Representan los datos y la estructura del árbol
- View: Interfaz JavaFX en Main.java - Presenta la información al usuario
- Controller: ArbolID3Visual.java - Contiene la lógica del algoritmo ID3

5.2. Componentes Principales

5.2.1. Clase Main

- Responsabilidad: Interfaz de usuario y control de la aplicación
- Funciones:
 - ✓ Inicialización de la GUI
 - ✓ Gestión de eventos de botones
 - ✓ Creación de datos de ejemplo
 - ✓ Configuración del layout

5.2.2. Clase ArbolID3Visual

- Responsabilidad: Implementación del algoritmo ID3 y visualización
- Funciones:

- ✓ Construcción recursiva del árbol
- ✓ Cálculo de entropía y ganancia de información
- ✓ Renderizado gráfico del árbol
- ✓ Clasificación de nuevos ejemplos

5.2.3. Clase Ejemplo

- Responsabilidad: Representación de datos de entrenamiento
- Funciones:
 - ✓ Almacenamiento de atributos
 - ✓ Gestión de clasificaciones
 - ✓ Acceso a datos específicos

5.2.4. Clase NodoID3

- Responsabilidad: Estructura de nodos del árbol
- Funciones:
 - ✓ Representación de nodos internos y hojas
 - ✓ Gestión de relaciones padre-hijo
 - ✓ Almacenamiento de atributos de decisión

6. ALGORITMO ID3 IMPLEMENTADO

6.1. Fundamentos Teóricos

El algoritmo ID3 construye árboles de decisión usando los conceptos de:

- Entropía: Medida de impureza de un conjunto de datos
- Ganancia de Información: Reducción de entropía después de dividir por un atributo
- Selección de Atributos: Elección del atributo con mayor ganancia de información

6.2. Proceso de Construcción

1. Cálculo de Entropía Inicial

$$H(S) = -\sum(p_i * \log_2(p_i))$$

2. Cálculo de Ganancia de Información

$$\text{Ganancia}(S,A) = H(S) - \sum((|S_v|/|S|) * H(S_v))$$

3. Selección del Mejor Atributo

- Se elige el atributo con mayor ganancia de información

4. División Recursiva

- Se crean subárboles para cada valor del atributo seleccionado

5. Condiciones de Parada

- Todos los ejemplos tienen la misma clasificación
- No quedan atributos disponibles

6.3.Características de la Implementación

- Manejo de Casos Base: Clasificación única y agotamiento de atributos
- Selección Óptima: Maximización de ganancia de información
- Estructura Recursiva: Construcción eficiente del árbol
- Clasificación Robusta: Manejo de valores desconocidos

7. DISEÑO DEL SOFTWARE

7.1.Diagrama de Clase

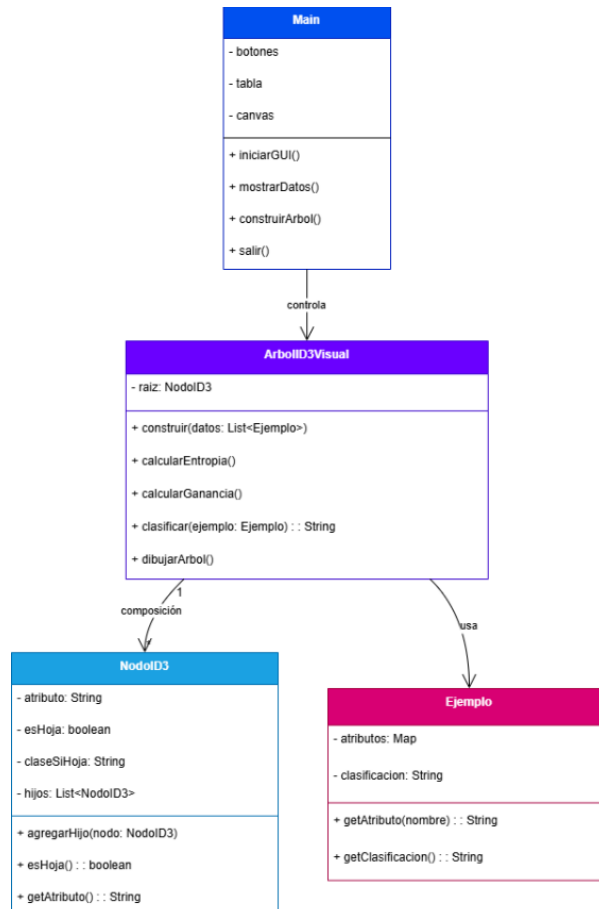


Imagen 1 Diagrama de clases

Fuente: Elaboración propia

7.2.Caso de Uso Principal

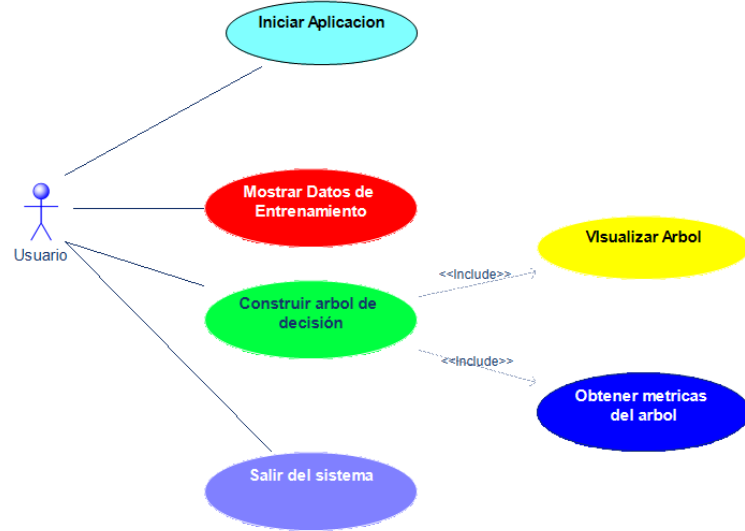


Imagen 2 Diagrama de casos de uso

Fuente: Elaboración propia

8. RESULTADOS ESPERADOS

8.1.Métricas de Calidad

- Precisión: Alta para datos de entrenamiento (100%)
- Generalización: Dependiente de la calidad y cantidad de datos
- Interpretabilidad: Excelente debido a la naturaleza del árbol

8.2.Beneficios del Sistema

- Automatización: Proceso de recomendación sin intervención manual
- Transparencia: Decisiones explicables y trazables
- Eficiencia: Clasificación rápida de nuevos casos
- Escalabilidad: Fácil incorporación de nuevos datos

9. CONCLUSIONES

9.1. Logros Alcanzados

- ✓ Implementación Exitosa: Algoritmo ID3 completamente funcional
- ✓ Interfaz Intuitiva: GUI clara y fácil de usar
- ✓ Visualización Efectiva: Representación gráfica del proceso de decisión
- ✓ Código Mantenible: Estructura orientada a objetos bien definida

7.2. Limitaciones Identificadas

- ✓ Conjunto de Datos Limitado: Solo 10 ejemplos de entrenamiento
- ✓ Atributos Categóricos: No maneja atributos numéricos continuos
- ✓ Sobreajuste Potencial: Árbol muy específico para datos de entrenamiento
- ✓ Validación Ausente: No implementa validación cruzada

7.3. Trabajos Futuros

- ✓ Ampliación de Datos: Incorporar más ejemplos de entrenamiento
- ✓ Interfaz Mejorada: Añadir formularios para entrada de nuevos casos
- ✓ Exportación: Permitir guardar y cargar árboles construidos

8. REFERENCIAS TÉCNICAS

8.1. Tecnologías Utilizadas

- Java 8+: Lenguaje de programación principal

- JavaFX: Framework para interfaz gráfica
- Collections Framework: Manejo de estructuras de datos

8.2. Patrones Implementados

- Patrón Recursivo: Construcción del árbol
- Patrón Observer: Eventos de botones
- Patrón Strategy: Cálculo de métricas

8.3. Estructuras de Datos

- HashMap: Almacenamiento de atributos y relaciones
- ArrayList: Listas de ejemplos y resultados
- HashSet: Conjuntos de atributos únicos

9. IMPLEMENTACIÓN

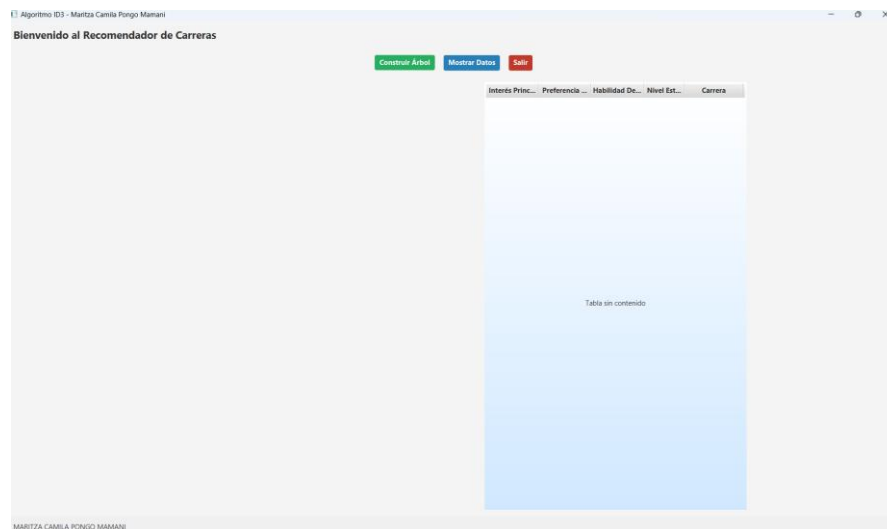


Imagen 3 - Bienvenida

Fuente: IntelliJ IDEA

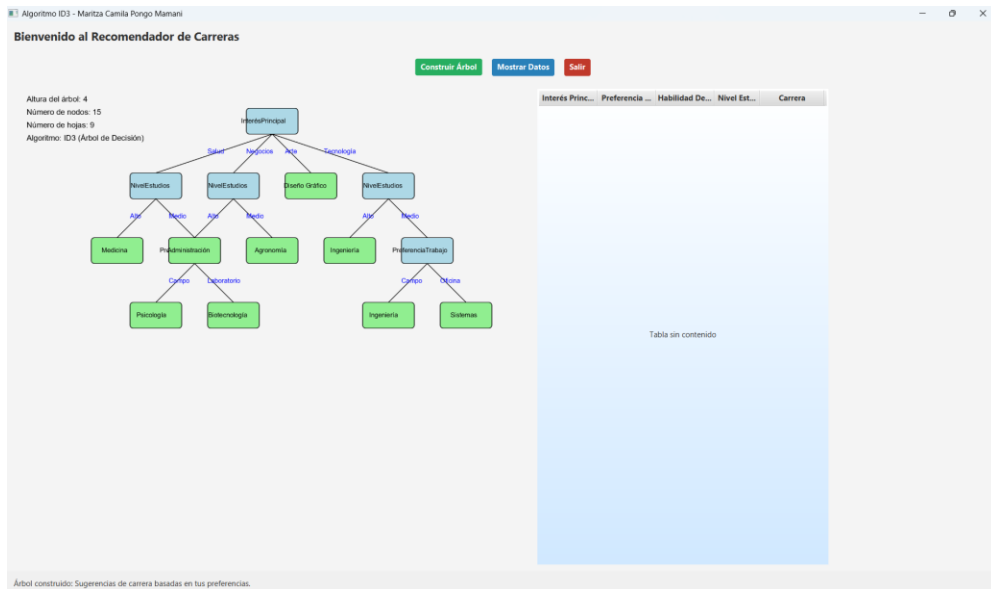


Imagen 4 - Construir Árbol

Fuente: IntelliJ IDEA

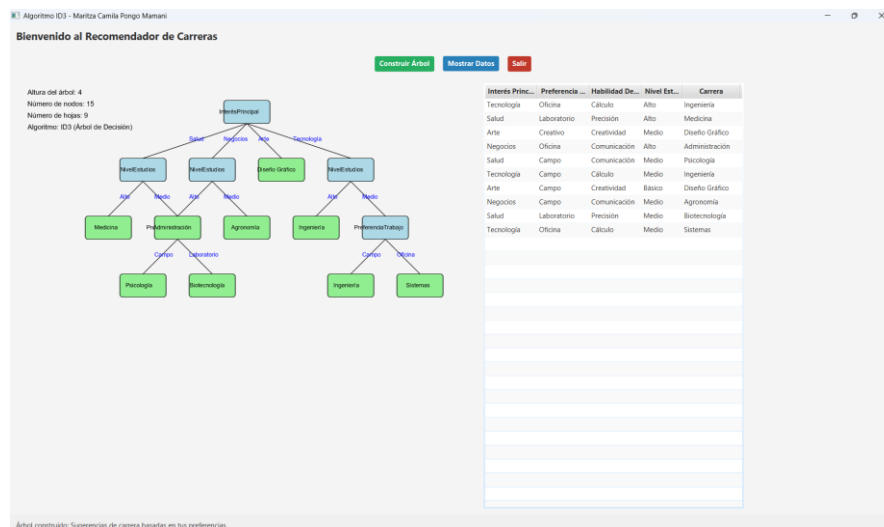


Imagen 5 - Mostrar Datos

Fuente: IntelliJ IDEA