**Aura Bot - OVA Machine Learning y Redes Neuronales**

Información del Proyecto

Nombre: Aura Bot

Tipo: Objeto Virtual de Aprendizaje (OVA)

Universidad: Universidad de Córdoba

Facultad: Educación y Ciencias Humanas

Programa: Licenciatura en Informática

Materia: Técnicas Avanzadas de Programación

Equipo de Desarrollo

Jesús David González Hernández - jgonzalezhernandez41@correo.unicordoba.edu.co

Yuliana Julián Otero Carmona - yoterocarmona20@correo.unicordoba.edu.co

Pedro Luis Romero Márquez - promeromarquez@correo.unicordoba.edu.co

Xiomara Hoyos Solano - xhoyossolano@unicordoba.edu.co

Enlaces

Repositorio: https://github.com/JESULIKE/Ova\_ML

Demo: https://ova-ml.vercel.app/

Descripción del Proyecto

Aura Bot es un Objeto Virtual de Aprendizaje interactivo diseñado para enseñar conceptos fundamentales de Machine Learning y Redes Neuronales. La plataforma ofrece una experiencia de aprendizaje progresiva y gamificada, donde los estudiantes pueden avanzar a través de diferentes módulos educativos con visualizaciones interactivas en tiempo real y demostraciones prácticas de algoritmos de clasificación.

Características Principales

Interfaz Moderna y Experiencia Inmersiva

Diseño Responsive: Adaptable a móviles y desktop con Vuetify

Efectos Visuales Avanzados: Elementos flotantes animados y transiciones suaves

Hero Sections Interactivas: Componentes con animaciones CSS3 y efectos de paralaje

Tema Personalizado: Gradientes dinámicos y glassmorphism

Sistema de Aprendizaje Interactivo

Demostraciones en Tiempo Real: Visualización de algoritmos de clasificación funcionando

Simulador de Entrenamiento: Demo interactivo que muestra el proceso de entrenamiento de modelos

Métricas Dinámicas: Visualización de precisión y épocas en tiempo real

Selección de Algoritmos: Interfaz para experimentar con diferentes algoritmos (Neural Networks, SVM, Random Forest, Logistic Regression)

Gamificación y Progreso

Sistema de Módulos Desbloqueables: Progreso basado en prerrequisitos

Indicadores Visuales: Barras de progreso, estadísticas y logros

Path de Aprendizaje Visual: Navegación intuitiva entre conceptos

Elementos Interactivos: Tooltips, hover effects y feedback inmediato

Stack Tecnológico

Frontend

Framework: Nuxt.js 3.15.4

UI Framework: Vuetify 3.8.9

JavaScript Framework: Vue.js 3.5.13

State Management: Pinia 0.11.0

Router: Vue Router 4.5.0

Icons: Material Design Icons (@mdi/font 7.4.47)

Herramientas de Desarrollo

CSS Preprocessor: Sass 1.89.2

Build Tool: Vite

Plugin: vite-plugin-vuetify 2.1.1

Persistencia: pinia-plugin-persistedstate 4.3.0

Estructura del Proyecto Actualizada

aura-bot/

├── .nuxt/ # Archivos generados por Nuxt

├── .output/ # Build de producción

├── .vscode/ # Configuración de VS Code

├── assets/ # Recursos estáticos

│ └── css/

│ └── main.css # Estilos principales

├── components/ # Componentes Vue reutilizables

│ ├── AlgorithmsSection.vue

│ ├── BasicsSection.vue

│ ├── DemoSection.vue # 🆕 Demo interactivo de clasificación

│ ├── HeroSection.vue

│ ├── HeroSectionIndex.vue # 🆕 Hero animado con elementos flotantes

│ ├── LearningPathSection.vue

│ ├── LearningTypesSection.vue

│ ├── ProgressCard.vue

│ └── RegressionClassificationSection.vue

├── composables/ # Composables de Vue

│ ├── answers.ts

│ ├── answersevaluacion.ts

│ ├── evaluacion.ts

│ └── quiz.ts

├── layouts/ # Layouts de aplicación

│ └── default.vue

├── pages/ # Páginas de la aplicación

│ ├── cuestionarios.vue

│ ├── evaluacion.vue

│ ├── index.vue

│ ├── ml\_index.vue # 🆕 Página principal ML con demos

│ ├── ml\_regressionvscalificacion.vue

│ ├── ml\_tiposdeaprendizaje.vue

│ ├── neuralnets.vue

│ └── quiz.vue

├── plugins/ # Plugins de Nuxt

│ ├── pinia.client.ts

│ └── vuetify.ts

├── server/ # API Server

├── stores/ # Stores de Pinia

└── [archivos de configuración]

Nuevas Funcionalidades Implementadas

1. Demo Interactivo de Clasificación (DemoSection.vue)

Características principales:

Visualización en Tiempo Real: Puntos de datos que se iluminan durante el entrenamiento

Selección de Algoritmos: Dropdown para elegir entre diferentes algoritmos de ML

Métricas Dinámicas: Muestra precisión y número de época en tiempo real

Frontera de Decisión Animada: Línea visual que separa las clases con efectos de pulso

Controles Interactivos: Botones para iniciar/detener el entrenamiento

Algoritmos Soportados:

Neural Networks

Support Vector Machine (SVM)

Random Forest

Logistic Regression

Código de ejemplo:

vue<template>

<div class="classification-demo">

<div class="demo-controls">

<v-btn @click="toggleTraining">

{{ isTraining ? 'Detener' : 'Entrenar' }} Modelo

</v-btn>

<v-select v-model="selectedAlgorithm" :items="algorithms" />

</div>

<div class="training-metrics" v-if="isTraining">

<span>Precisión: {{ trainingAccuracy }}%</span>

<span>Época: {{ currentEpoch }}</span>

</div>

</div>

</template>

2. Hero Section Avanzado (HeroSectionIndex.vue)

Características principales:

Elementos Flotantes Dinámicos: 20 elementos con iconos que flotan con diferentes velocidades

Animaciones CSS Avanzadas: Bounce, pulse, gradient y sparkle effects

Path de Aprendizaje Visual: Nodos interactivos que muestran la progresión

Estadísticas Animadas: Cards con métricas del curso (horas, módulos, certificados)

Responsive Design: Adaptación completa a dispositivos móviles

Efectos Visuales:

Gradientes dinámicos de fondo

Elementos con glassmorphism

Animaciones de entrada escalonadas

Efectos de hover interactivos

3. Página ML Index Renovada (ml\_index.vue)

Estructura mejorada:

vue<template>

<v-app>

<v-main class="ml-bg">

<section class="hero-section">

<HeroSection @scroll-to-content="scrollToContent" />

</section>

<section class="demo-section">

<DemoSection />

</section>

<section class="basics-section">

<BasicsSection />

</section>

</v-main>

</v-app>

</template>

Estilos y Animaciones Avanzadas

Animaciones CSS Personalizadas

Floating Elements:

css@keyframes float {

0%, 100% { transform: translateY(0px) rotate(0deg); }

50% { transform: translateY(-20px) rotate(180deg); }

}

Training Animation:

css.data-point.training {

transform: scale(1.3);

box-shadow: 0 0 20px currentColor;

}

Decision Boundary:

css.decision-boundary.active {

background: #8b5cf6;

box-shadow: 0 0 20px #8b5cf6;

animation: boundaryPulse 2s infinite;

}

Efectos Visuales

Glassmorphism: backdrop-filter: blur(10px) con transparencias

Gradientes Dinámicos: Transiciones suaves entre colores

Box Shadows: Efectos de profundidad y elevación

Transform Animations: Escalado, rotación y traslación

Módulos Educativos (Actualizado)

Módulo 1: Machine Learning Interactivo

Duración: 2-3 horas

Nivel: Principiante

Nuevas características:

Demo interactivo de clasificación

Visualización de algoritmos en tiempo real

Selección dinámica de algoritmos

Métricas de entrenamiento en vivo

Contenido Mejorado:

¿Qué es Machine Learning? - Con visualizaciones interactivas

Tipos de Aprendizaje - Demostraciones prácticas

Regresión vs Clasificación - Simulador integrado

Evaluación de Modelos - Métricas en tiempo real

Instalación y Configuración (Actualizada)

Prerrequisitos

Node.js (versión 18 o superior)

npm o yarn

Git

Pasos de Instalación

Clonar el Repositorio:

bashgit clone https://idx.google.com/jeyupe-4893800

cd aura-bot

Instalar Dependencias:

bashnpm install

# o

yarn install

Ejecutar en Modo Desarrollo:

bashnpm run dev

# o

yarn dev

Construir para Producción:

bashnpm run build

# o

yarn build

Configuración de Componentes

DemoSection.vue - Configuración

Props disponibles:

javascriptdefineProps({

demoAlgorithms: {

type: Array,

default: () => ['Neural Networks', 'SVM', 'Random Forest', 'Logistic Regression']

}

})

Estados reactivos:

javascriptconst isTraining = ref(false)

const selectedAlgorithm = ref('Neural Networks')

const trainingPoints = ref([])

const trainingAccuracy = ref(0)

const currentEpoch = ref(0)

HeroSectionIndex.vue - Configuración

Elementos configurables:

javascriptconst icons = ['🧠', '🤖', '⚡', '🔬', '🌟', '💡', '🎯', '🚀']

const colors = ['#ff6b6b', '#4ecdc4', '#45b7d1', '#96ceb4', '#feca57', '#ff9ff3', '#54a0ff', '#5f27cd']

const pathNodes = [

{ icon: '🎯', label: 'INICIO', tooltip: '¡Empezamos aquí!' },

{ icon: '🧠', label: 'ML BÁSICO', tooltip: 'Aprende los básicos' },

{ icon: '🕸️', label: 'REDES', tooltip: 'Redes neuronales' },

{ icon: '🚀', label: 'EXPERTO', tooltip: '¡Eres un experto!' }

]

📱 Responsive Design Mejorado

Breakpoints Implementados

Desktop (960px+):

Layout completo con sidebar

Animaciones avanzadas

Todos los efectos visuales

Tablet (600px - 960px):

Layout adaptativo

Componentes reorganizados

Efectos simplificados

Mobile (< 600px):

Layout vertical

Navegación colapsada

Optimización táctil

Media Queries Clave

css@media (max-width: 960px) {

.hero-title { font-size: 2.5rem; }

.learning-path { flex-direction: row; flex-wrap: wrap; }

}

@media (max-width: 600px) {

.hero-title { font-size: 2rem; }

.stat-item { padding: 0.8rem 1rem; }

}

🎯 Mejoras en la Experiencia de Usuario

Interactividad Avanzada

Tooltips Informativos: Explicaciones contextuales

Hover Effects: Feedback visual inmediato

Loading States: Indicadores durante procesos

Error Handling: Manejo elegante de errores

Contacto

Email: jgonzalezhernandez41@correo.unicordoba.edu.co

Universidad: Universidad de Córdoba

Facultad: Educación y Ciencias Humanas

Logros del Proyecto

✅ Interfaz Moderna e Interactiva

✅ Demostraciones en Tiempo Real

✅ Sistema de Progreso Gamificado

✅ Responsive Design Completo

✅ Animaciones CSS Avanzadas

✅ Arquitectura Escalable

Última actualización: 18 Junio 2025 - Versión 2.0 con componentes interactivos