

ANDREA MARGARITA MALDONADO MORALES

| am.maldonado.morales04@gmail.com | [Linkedin](#) | [Github](#) |

Benemerita Universidad Autonoma de Puebla (BUAP)

Puebla City, Mexico

Major: Information Technology Engineering

Expected Graduation date: May 2027

- **Relevant Courses:** Relevant Courses: Databases, Operating Systems, Software Engineering, Algorithms, Data Structures, Web Development, OOP, Programming, Computer Networks, Calculus, Linear Algebra

SKILLS

- **Programming languages:** C++ (proficient), Python (proficient), JavaScript (basic), , SQL, PHP (Basic).
- **Web development:** HTML/CSS, Astro , responsive design, cross-browser compatibility, agile development.
- **Mobile development:** Kotlin
- **Backend development:** Django REST Framework, FastAPI, Node.js, Express.js, PostgreSQL, MongoDB
- **Software and tools:** Figma, Git, VS Code, Android Studio, Bash, Linux, Grafana

PROJECTS

Custom Presentation Web Platform

Role: Full-Stack Developer

- Spearheaded the development of a web-based presentation tool for creating, editing, and publishing slideshows locally or on a private server.
- Designed the initial UI/UX layout for the presentation web editor utilizing Figma before engineering the frontend components with React and Angular.
- Deployed the platform on a Linux server, establishing secure remote administration, configuration, and monitoring via Linux-to-Linux SSH connections.
- Configured local hosting and utilized Cloudflare Tunnels to expose secure ports, enabling multiple users to reliably connect, collaborate, and modify presentations from any network.
UI/UX Engineering: Designed and programmed over 20 reusable visual components from Figma to React and Angular, standardizing the platform's design system and accelerating frontend development.
Infrastructure & Security: Configured secure Cloudflare Tunnels and Linux-to-Linux SSH connections, enabling seamless remote access and real-time collaboration for a team of 2 developers without relying on public servers.
Development & Deployment: Spearheaded the end-to-end development and local Linux server deployment of a web-based presentation tool for creating, editing, and publishing slideshows.

Meta Glass Sales & Communication Agent (Hackathon)

Role: Backend Architect

- Designed and implemented a robust backend architecture for an immediate communication and sales application integrated with Meta Glass hardware.
- Developed an intelligent agent system that processes incoming voice calls to extract key data points and routes the information to a Model Context Protocol (MCP).
- Orchestrated a multi-agent workflow to dynamically retrieve, analyze, and store contextual data (e.g., pricing, sales data, pet information), ensuring seamless continuation of transactions upon user callbacks.
- System Architecture: Designed, integrated, and deployed a fully functional backend utilizing the Model Context Protocol (MCP) within a strict 12 hour hackathon timeframe.
- Multi-Agent Orchestration: Architected a dynamic workflow of 4 simultaneous intelligent agents capable of processing live voice calls to extract 50 key data points (e.g., pricing, pet information) in real time.
- Contextual Data Management: Built a robust data routing system that seamlessly retrieves, analyzes, and stores context, ensuring flawless continuation of sales transactions upon user callbacks.

Meta Glass Real-Time Accessibility Assistant

Role: Computer Vision Developer

- Developed a real-time visual assistance solution for the visually impaired, functioning as an active "eyes" system through Meta Glass.
- Implemented lightweight computer vision models utilizing Python, OpenCV, and YOLO to optimize processing power and avoid massive infrastructure dependencies within the smart glasses' hardware constraints.
- DEngineered visual processing pipelines to analyze live video feeds for critical navigation, such as reading traffic lights for safe crossing and identifying product sections in supermarkets, outputting automated voice guidance.
- Hardware Optimization: Engineered a data partitioning (chunking) strategy to bypass the Meta Glass 1 FPS hardware transmission limit, successfully feeding the YOLO model the required 2+ FPS to guarantee accurate, real-time object recognition.
- Computer Vision Pipeline: Implemented lightweight computer vision models using Python, OpenCV, and YOLO to optimize processing power within severe smart glass hardware constraints.
- Real-Time Navigation: Developed visual processing pipelines to analyze live video feeds for critical navigation (e.g., reading traffic lights, identifying supermarket sections), outputting automated, low-latency voice guidance for visually impaired users.

FiTPet

Role: Android Developer

- *FitPet is an Android mobile application developed using Kotlin and Jetpack Compose that gamifies physical activity through a virtual pet experience. Inspired by a Tamagotchi-style concept, the app motivates users to stay active by tracking daily steps using built-in device sensors and translating that activity into the pet's growth, mood, and evolution.*
- *The project follows the MVVM architectural pattern to ensure a clean separation of concerns and uses reactive state management to update the user interface in real time.*
- *It features a modular UI built with reusable components and implements core logic for goal tracking, progress visualization, and dynamic feedback based on user activity.*

AI-Driven Pac-Man

Role: AI & Game Developer

- *Artificial Intelligence: Engineered a multi-tiered AI system for 4 enemy NPCs, implementing A* (A-Star) and IDA* pathfinding algorithms for strategic targeting, alongside Alpha-Beta pruning for cooperative, multi-agent trapping behaviors.*
- *Graphics & Rendering: Built a custom 2D rendering engine utilizing Python, Pygame, and PyOpenGL, efficiently mapping BMP sprite textures and rendering the game environment using orthographic projection (gluOrtho2D).*
- *Game State & Movement Logic: Developed a precise grid-based collision system by parsing CSV map data into a 10x10 Control Matrix, translating exact pixel coordinates to matrix intersections to ensure flawless directional routing and player input handling.*