Daniel Aguila

dbaguilasifuentes@utexas.edu | (713) 705-2197 | US Citizen
danielaguila.dev | linkedin.com/in/aguiladaniel | github.com/Daniel-Aguila

EDUCATION

The University of Texas at Austin, Austin, TX — May 2027
Master of Science in Computer Science
Relevant Coursework: Advanced Operating Systems, Parallel Systems, Virtualization, Android Programming

University of Houston, Houston, TX — May 2021
Bachelor of Science in Computer Science

EXPERIENCE

Atser Systems, Houston, TX — December 2024 – Present

Junior Software Developer

- Defined the socket/REST interface for an ESP32 scale dongle and mentored its engineer on integration.
- Led 1:1 client sessions to scope requirements and coordinate delivery of the scale integration.
- Built VoiceIT (Go, Azure OpenAI): voice and image input auto-fills test forms, cutting entry from 30+ fields to 10.
- Managed OpenAI coding agents to build VoiceIT's backend APIs, reviewing and refining generated code.
- Reduced cold-start time and improved scaling stability by refactoring containers and tuning CPU/memory limits.
- Instrumented Python dashboards (NumPy/Matplotlib) to monitor sync errors during an AWS to Azure migration.
- Shipped React Native offline-first sync and storage for a construction app used across 150+ devices.
- Leading migration to native Android (Kotlin, Jetpack Compose) to improve performance and reduce third-party dependency risk.

Rice University – Aiden Lab, Houston, TX — July 2020 – May 2021

Machine Learning Intern

- Improved cancer-gene detection recall by 15.4% by implementing a preprocessing pipeline for genomic Hi-C data.
- Trained and evaluated U-Net++ models in PyTorch, presenting results in lab research reviews.

PROJECTS

- **STM32 LED** (C, STM32) **–** Configured GPIO and the system clock to blink an LED using timer-based delays.
- **Linux Container Runtime** (C) – Isolated processes with user, PID, mount, UTS, and network namespaces; sandboxed them by dropping all capabilities and installing a seccomp-BPF syscall filter.
- **Virtio Device Backend** (C) – Implemented the virtio/vhost-user virtqueue backend for a team-built a VM monitor.
- **GPU-Accelerated K-Means** (CUDA C++) – Parallelized clustering with warp/block kernels and shared memory.
- **Pintos File System** (C) – Built multi-level indexed inodes with sparse file support, FIFO ticket locks for reader-writer fairness, and a stat syscall.
- **Unix Shell** (C, POSIX) – Implemented parsing, pipelines, and process control with fork/exec/wait.
- **Two-Level Parallel Prefix Scan** (C++) – Implemented a two-level scan synchronized with custom spin barriers.
- **Capybara OS** (Rust) – Bare-metal x86-64 hobby kernel (in progress).

OPEN-SOURCE CONTRIBUTIONS

Avogadro2 (C++/Qt)

- Added backend support for in-app molecule creation in the Molecules list (avogadroapp #509).
- Fixed File > Close to close only the active molecule instead of the whole app, and renamed the action "Close Active Molecule" (avogadrolibs #2002).

SKILLS

Languages: C, C++, Python, Rust, Go, Kotlin, TypeScript, SQL
Systems & Tools: Linux (namespaces, cgroups, seccomp), CUDA, Thrust, Nsight, QEMU, Git, STM32
ML & Data: PyTorch, NumPy, Pandas, Matplotlib
Mobile & Cloud: React Native, Jetpack Compose, Android SDK, Gradle, AWS, Azure, PostgreSQL, SQLite
Spoken: English (Native), Spanish (Native), Japanese (Intermediate), Korean (Conversational)