Emanuel Cortes Lugo

Software Developer & Accessibility Engineer

Experience

Application Programmer 2

University of Central Florida | Orlando, FL | Dec 2022 - Present

- Full Stack development
- Leads projects (e.g., taskforce/AI initiatives)
- Designs data pipelines and analytics reporting
- Builds and maintains production services and internal tools for academic programs
- Provides client support
- Mobile development
- Implements integrations with Canvas LMS/LTI and backend automation scripts (Python/Django/React)

Techranger

University of Central Florida | Orlando, FL | Dec 2020 - Dec 2022

- Full Stack development
- Developed functionality that provides accessibility
- Developed new features for existing repositories
- Added/updated DB functionalities
- Provided code guidance, best practices, and data wrangling to new Techrangers

Avionics Instructor

Avionics USA | Orlando, FL | Aug 2019 - May 2021

- Educated students on the proper way of installing equipment using a simulation
- Displayed radios and sensor components using a simulation on a mobile device
- Taught students how to calculate voltage and current
- Taught students about electronic logic
- Educated students on installing electrical components: GMA340, GTR200, GTX345, and Encoders
- Built and troubleshoot wire harnesses
- Instructed students properly to communicate with the control tower over the radio

Education

University of Central Florida

M.S. in Machine Learning
(Expected Completion 2026)

University of Central Florida

B.S. in Computer Engineering
(August 2022)

Skills

Git/Hub/Lab | AWS | Dev-Ops | LLM
Debugging | Img Processing | Tech Comms | Spanish
XGBoost | BERT

Programming Languages

Proficient in Python | Proficient in JavaScript | Proficient in PHP

Community Engagement

Teaching and Learning with AI Conference

Orlando, FL | July 2024 & May 2025

- Participated in UCF's annual AI conference aimed at helping educators leverage AI in the classroom
- Provided demos on GPT-4o and ChatGPT and served as a staff volunteer

AR/VR Innovation Discovery (AVID) Events

Orlando, FL | 2023 - 2025

- Committee member, event hosted by UCF's Faculty Multimedia Center and Pegasus Innovation Lab
- These events build a community of professionals interested in virtual and augmented reality, sharing real-world use cases, research, and interactive demonstrations

Open-Source Community

- Active on GitHub and UCF Open Slack, providing support to other institutions adopting UDOIT and contributing bug reports and documentation updates

Profile

Adaptable software developer and accessibility specialist with experience in full-stack web development and digital accessibility. Proven track record building and maintaining open-source tools and supporting instructional content at the University of Central Florida. Expertise spans PHP/Symfony, React, and Python/Django, alongside strong knowledge of accessibility standards and captioning workflows. Active participant in university initiatives exploring artificial intelligence (AI) and extended reality (XR) technologies and passionate about inclusive design. Pursuing a deeper understanding of how ML systems work by obtaining a Master's degree.

321-805-9863 emanuelcodes@gmail.com LinkedIn github.com/emanuelGitCodes

Projects

UCF Here

- Lead the development of *UCF Here 2.0*
- Planned and restructured the UI
  - UI was developed using a **Single Page Application** with React
  - Restructured the UI in a way that was more UX-friendly
- **Increased the speed of the backend by 9x**
  - Updated functions to make use of Django ORM
  - Used **GraphQL** to fetch the exact data needed from LMS API calls
- Implemented **Celery** to manage asynchronous tasks
  - Implemented tasks in a way that makes the user experience feel seamless
  - Added **Cron Jobs** to handle daily tasks off peak hours
- Created a new deployment process to run the application on an **AWS EC2** instance
- Updated and ran migrations to accommodate the new changes
- Implement **NGINX** as a reverse proxy server
- Made use of **UWSGI** to deploy dynamic content
- Created Dockerize the application for different environments, such as Local, QA, or Production
- Set up *LTI 1.0* and had it connect to the LMS Canvas
- Optimized the number of query calls done to the DB by removing n+1 error
- Added a check to update DB when professors merged courses in the LMS
  - *Updated all section data to keep up with the latest content coming from the LMS*
- Based on changes done in a course LMS roster, updated the DB to show the up-to-date list of students in minutes
- Updated the application to make use of *multiple databases*
  - Updated main application database and connected to secondary database for data analytics
- Developed the **data analytics** functions to extract information such as:
  - *How many colleges have made use of the application?*
  - *Who are the professors who have used the application most?*
  - *How many courses are currently making use of the application?*
    - *Has the number grown compared to previous semesters?*
- Deployed mobile applications to the *Google PlayStore* and *Apple App Store*
- Reviewed pull requests
  - Provided constructive feedback to educate collaborators on what best practices are
- Communicated with clients
  - Went over any issues faculty had
  - Educated faculty on the best way to make use of the application
  - Met with students who had issues with the mobile version and looked for a solution
- Improved the way **Sentry** managed traces to reduce Sentry cost
  - Stopped the reporting of redundant traces such as ELBHealthCheck from AWS
- Provided better error handling by using proper exceptions such as the ones that come with Canvas API

Projects

Password Changer

- Diagnosed the cause for sending **error messages** when the password was successfully updated
- Updated the code repo to display the correct information to the user
- *Increased code-base security by implementing a method in the **Django** code that eliminated storing user passwords in the log history*
- Added a new feature allowing users to choose and update multiple canvas environments' passwords
    - Updated the frontend code to allow fetching and displaying multiple canvas environments
    - Added a **security method** that only allows the updating of passwords if one or more canvas environments are chosen
    - *Altered existing API calls* to allow for parsing new data between the client and server sides
    - Backend functions were updated to process multiple user choices
- Wrote **unit testing** to confirm the deletion of the users' password from the logs
- Wrote **unit testing** to confirm that updating multiple canvas environments proceeded as intended

Code Correction Transformer - C++ Error Diagnosis & Auto-Fix

Two-stage transformer pipeline to classify error types from compiler/runtime output and generate fixes for C++ submissions.

- **Stage 1 (Classification):** Fine-tuned **CodeBERT** for multi-class error classification using a curated C++ subset (**~**39.9k train / 19.1k test). Implemented custom tokenization, class label mapping, and detailed metrics for each label. Produced full **confusion matrix** and per-class precision/recall/F1 for *Runtime Error, Time Limit Exceeded, Memory Limit Exceeded;* overall test **accuracy ~77.4%** with strong recall on *Runtime Error.* Saved model + tokenizer artifacts for reuse.
- **Stage 2 (Generation):** Trained **CodeT5 seq2seq** to propose code fixes conditioned on source + stderr prompts. Used generation with max length and evaluation on held-out set (reported **eval_loss ~0.0568**), with optional **BLEU** for sequence quality tracking.
- **Infra & Efficiency:** Executed on **A100 40GB** (Colab) with bfloat16/mixed precision, gradient accumulation, and data collators. Wrote deterministic seeds and logging. Persisted checkpoints and confusion matrix.
- **Impact:** This project showcases comprehensive ML ownership, encompassing dataset preparation, training, evaluation, and artifact generation. The techniques demonstrated can be directly applied to media data science, including diagnosing platform errors, automating remediation suggestions, and developing assistive tools for engineers.

Tech: Python, PyTorch, Hugging Face (Trainer/Seq2SeqTrainer), scikit-learn metrics, Matplotlib; GPU acceleration (CUDA)

Projects

DNABERT-Style Genome Modeling - Custom k-mer Tokenizer & BERT

Standalone module implementing a DNABERT-style workflow for DNA sequences with ML training/evaluation utilities.

- **Tokenizer & Data:** Built a **K-mer tokenizer** for DNA, FASTA parsing, and robust preprocessing (cleaning, stride-based chunking, max/min lengths). Generates K-mer text suitable for BERT tokenization; batched encodes to tensors.
- **Modeling:** Constructs a BERT-family config tailored for genomic tokens and instantiates **BertForSequenceClassification/BertForMaskedLM**. Supports **masked language modeling (MLM)** pretraining with curriculum masking (gradually increasing mask probability).
- **Training Loop:** Custom **AdamW** optimizer with cosine LR warm-up/schedule, gradient clipping, mixed precision, checkpointing, early stopping, and confusion-matrix plotting. Memory-optimized dataloaders.
- **Impact:** Shows capability to create **domain-specific foundation models** and pipelines from scratch. Skills transfer to text/logs/sequences in media (e.g., session/token sequences, subtitle streams, anomaly sequences) and to **generative** or **representation-learning** tasks.

Tech: Python, PyTorch, Hugging Face configs/models, BioPython, NumPy, scikit-learn metrics, Matplotlib/Seaborn

Vein Scanner

- Machine Learning application that allows for a user's Log In and Sign Up based on their hand vein patterns.
- Detect and recognize palm and individual fingers utilizing the Python library MediaPipe.
- Input is determined by how many fingers the user has extended.
- Implement a security feature that only allows people to sign up by scanning a specific QR code.
- Utilizing an infrared sensor and IR light, images of the hand vascular vein pattern are captured and processed.
- Recognition of users is achieved using a **one-shot image recognition siamese neural network**.
- Stores user images on a database.
- Implement Python multi-threaded library to run multiple functions asynchronously.

Projects

Senior PHP / Symfony Engineer – UDOIT 3 Accessibility Scanning Platform

Re-architected the open-source **UDOIT** tool to perform large-scale, parallel accessibility scans of Canvas LMS courses, driving **40% faster throughput and real-time progress feedback** for instructors.

- Introduced **Symfony Messenger** transports and supervisor-managed consumers.
- Create bash file configurations to enable Symfony Messenger to run **asynchronous** tasks.
- Designed three message types (FullRescan, ScanContentItem, FinishRescan) and corresponding handlers to break a scanning of a course into hundreds of independent units processed in parallel.
- Implemented smart batching and metadata-flag workflow to reliably detect the actual last worker without Redis or DB locks.
- Scaled from single-threaded scans to n parallel threads.
- Added leaky-bucket back-off for Canvas API rate limits so parallel tasks wait until the API becomes available.
- Traced and fixed Doctrine proxy errors (EntityNotFound, cascade persist) by normalizing entity life-cycle inside handlers.
- Patched missing flushes so thousands of Issue rows are now persisted deterministically after every scan.
- Mapped JSON results to UDOIT domain objects, added rule-level severity mapping and custom ignore makers.
- Docker-Compose stack (NGINX, PHP-FPM, MySQL, Node micro-service).
- Authored reusable supervisor templates parameterized by environment variable.
- Wrote comprehensive README and PR templates; documented new message bus flow.
- Added toast/Web-Socket notifications when FinishRescanHandler flips the report ready flag.
- **Impact:** Increase speed for the client by 40%; platform supports future multi-threaded tasks; able to add new Docker containers running on separate threads