Eidan J. Rosado, Ph.D.
Teaching Statement

Teaching Philosophy

My teaching philosophy is grounded in the belief that computing education should develop technical skill, systems thinking, and reflective awareness. As an educator and a major proponent of responsible AI, I aim to help students not only build effective software and intelligent systems but also understand their impact on individuals from psychological, social, and ethical viewpoints. I treat AI fundamentals and responsible AI not as electives bolted onto a traditional curriculum, but as essential preparation for anyone building or deploying intelligent systems today. My approach is shaped by my background in software engineering, embodied and agentic AI research, immersive training environments, and computational psychology, which allows me to connect algorithmic reasoning with human factors, design ethics, and responsible innovation.

I believe students learn best when they can see how computation applies to real-world situations. Whether I am teaching application development, cloud and serverless architecture, AI fundamentals, or agentic AI workflows, I emphasize active experimentation, iterative design, and critical inquiry. In AI-focused instruction, I start with fundamentals (how models learn and fail, what data represents, and where human judgment still matters) before moving to applied topics such as retrieval, agents, and deployment. My goal is for students to leave my courses with the ability to build systems that are production-ready (tested, observable, and maintainable) and to question how those systems interact with users, data, and society. Teaching computer science, for me, means shaping ethical engineers and computational thinkers, not merely instructing students on how to code.

Teaching Experience and Approach

I have taught courses and modules across the following areas:

- Software development
- Software testing and automation
- AI fundamentals
- Responsible AI

Across these courses, I integrate professional engineering principles with adaptive learning techniques. My teaching reflects my industry work: rapid prototyping and critical thinking at the core. In development classes, I focus on modular design, test-driven development, and continuous integration. In AI fundamentals and responsible AI classes, I teach how models learn and fail, how to evaluate outputs, how data quality shapes behavior, and how to design systems with clear safety boundaries and accountability from the start. Students build real-world applications ranging from prototypes to production-grade architectures while learning version control, test automation, and debugging strategies aligned with industry standards. I emphasize learning through building, with reflective practice and peer code reviews to strengthen both collaboration and technical rigor.

In addition to academic teaching, I conduct internal workshops and mentorship sessions on re-

sponsible AI, production ML systems, and agentic development practices, extending the same fundamentals-first, ethics-integrated approach I use in the classroom. These sessions connect theoretical understanding with the practical realities of engineering data pipelines, retrieval systems, and AI-assisted software workflows, including how data flows, how failure modes emerge, and how models and agents should be governed in production. I often use case-based learning: scenarios where a recommendation system, dialogue agent, or embodied platform must balance accuracy, safety, fairness, and interpretability. These experiences reinforce my conviction that AI fundamentals, AI literacy, platform engineering, and responsible design should be core competencies in every computing curriculum.

As an author of the Cloud Bytes Collection and *Engineering Best Practices for the Agentic Developer*, I also work to make complex technical material accessible beyond the classroom, translating abstract systems concepts into guidance practitioners can apply immediately.

Teaching Methods and Pedagogy

My pedagogy draws from project-based, experiential, and interdisciplinary learning. I design courses and supplemental material around three principles:

1. **Scaffolded Technical Complexity:** Students progress from guided labs to open-ended projects that mirror real engineering challenges. An app prototype becomes the foundation for testing pipelines, API integration, deployment, and ultimately data-driven or agentic functionality.

2. **Ethical and Reflective Integration:** Each technical milestone is accompanied by reflection on design choices, user impact, and data ethics. In AI-related instruction, students build a foundation in responsible AI (evaluating model outputs, identifying risk, and documenting trade-offs) alongside the technical work so they look beyond implementation alone.

3. **Feedback as Iteration:** I emphasize iterative improvement in both code and understanding. Peer reviews, instructor feedback loops, and continuous integration checks mirror the reflective cycles found in professional practice.

Future Teaching Goals

Looking ahead, I plan to expand and formalize courses that reflect my dual expertise in agentic systems engineering and behavioral research: courses where students build dialogue agents or embodied AI prototypes while grappling with how design choices shape machine behavior and human experience. I will continue developing dedicated modules in AI fundamentals and responsible AI, covering model evaluation, prompt and agent design, data stewardship, safety boundaries, and the social implications of deployed systems, and advocate for their inclusion as standard requirements rather than optional add-ons. I envision labs that span cloud-native development, edge deployment, and simulation, helping students understand intelligent systems as full stacks, not isolated models.

Ultimately, my teaching goal is to inspire students to see themselves not merely as coders, but as creators of meaningful, responsible systems. Whether through apps, agentic architectures, robotics platforms, or large-scale analytics, I want students to internalize that designing technology is an act of empathy and foresight. Exceptional engineers and educators share a common skill: designing for both how something works and how it is experienced.