Rafael Castro Gonçalves Silva

Professional Cover Letter

I'm a working computer scientist and a developer.

Things that I like and care

I like challenges in programming! But not *fake challenges* like understanding frameworks, though it perhaps is one of the most valued things in the industry nowadays. I like to reason about the semantics of the programming language or stuff like "what is the time complexity of this function?" Real chanllenges in programming usualy requires you to be creative and *educate yourself* about it. There is no silver bullet, you need to learn about the thing that your solving, you need to read papers and about it! It doesn't matter which framework you are using if the problem NP-Hard or if the size of data is so huge that SQL databases can't hadle it.
People should notice that what makes someone a good programmer is to undestand Computing Science foundations. Yes, tools are important to be more efficient and so on. And engineering is about using those tools and the resources that you have to solve some problem. But a tool just *stay* important for some time, eventualy some other new tool appears to take its places. So in this section I will only talk about ideas ands values that I like and care, but not the tools that I use.
I care about code quality, test coverage, static analysis, and well-written code. Tests are not the ultimate tool in software quality, but it is the thing that the industry has widely adopted. I think we should not forget other styles of quality insurance - I mean, there is a huge and diverse academic research field about that. If you are dealing with a distributed system why not model it in a model checker?
I think that software development should follow some basic "practical software engineering", like CI/CD, code review, git flow and so on. Comunication in programming is also very importat, so we should always make clear "why some piece of code exists", other questions usualy are easy to answer by just reading the thing. In my opinion, undestanding the goal of some function is more important than undestanding how it does things.

How I'm like

I'm nerd, progressist and I like dogs.
I'm a quick learner and learning is the most important thing for me, so I wouldn't like to work in a place where decisions are made based only on using the technologies that the employees already know. Developers should be eager to learn the best technology for this specific problem. I rather work in place where I'm continuasly learning,

rafaelcgs10@gmail.com