Resumen de:

https://medium.com/@colinritman/the-philosophy-of-debugging-life-why-your-first-commit-should-always-be-terrible-f09cb256efa4

In the late 1800s, philosopher Charles Sanders Peirce developed a concept called fallibilism. We learn not by avoiding mistakes, but by making them systematically and learning from each iteration. Junior developers often agonize over writing “clean code” from day one. Senior developers know that clean code is refactored code. They write the messy version first, because they understand that perfection isn’t a starting point — it’s a destination you reach through iteration.. Every bug report becomes data. Every feature request reveals assumptions you didn't know you had. Security vulnerabilities surface. Edge cases emerge. Performance bottlenecks appear. In coding terms, this is the difference between architecture astronauts who design perfect s and those who ship a minimum viable product. You understand your users by shipping aminimum viable product and watching how they actually use it. This maps perfectly onto the modern software development lifecycle. You don't understand your user by reading personas and user stories (though these help). You understand them by shipping an effective product. This is what separates junior developers from senior ones. It's what makes DevOps culture different from traditional software development. It is a process of moving from a problematic situation to a resolved one through experimental action. In DevOps, we learn by doing and observing the results of our work. We don't learn by thinking harder — we learn from doing. In Dewey's framework, knowledge isn't something static that we possess; it's something dynamic that we create.