# The Many Career Paths of a Software Engineer

Universidad de Oviedo - 2025

Miguel Fernández Fernández

the signal
independent
software engineers

# In an average scenario, you will spend the next 40 years doing *this*

Daniel Gayo Avello (circa 2003)

# Before going any further…

## Miguel Fernández

### Maker

I am focused on developing my own products and converting them into businesses. This allows me to experiment with new ideas and push the boundaries of my own creativity and knowledge.

This is important to me because I have an ambition for **freedom** and **control** —to shape my own path and create opportunities for growth on my own terms. At the same time, I am deeply passionate about understanding users' problems and crafting solutions that are both effective and simple, ensuring that the value I deliver is not only innovative but also genuinely meaningful to those I serve.

### Consultant

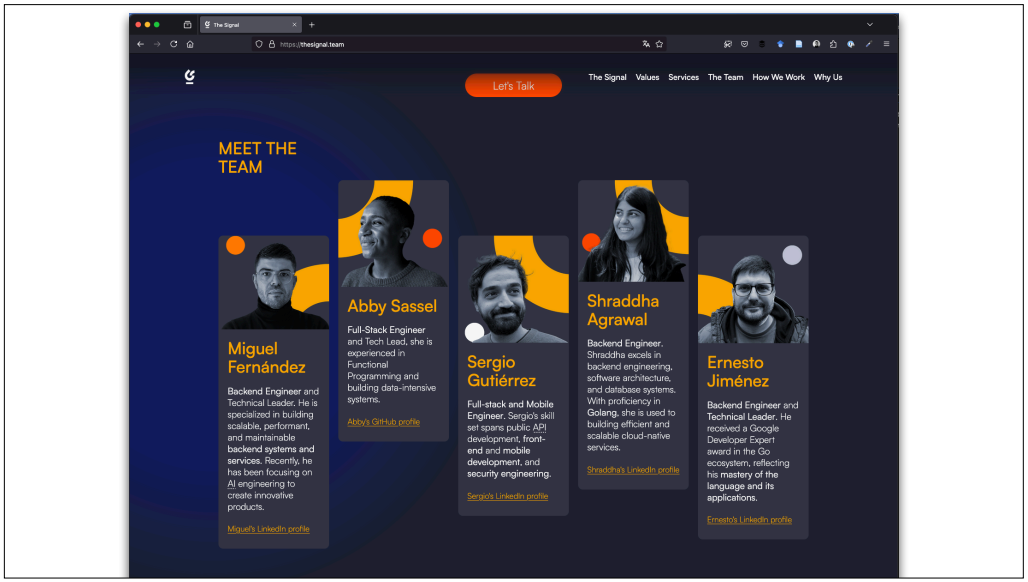
As part of The Signal Team, I collaborate with teams and individuals to unlock the potential of AI in their projects. This involves demystifying complex technologies, integrating AI into existing systems, and building reliable backends and systems from inception all the way up to production.

### Engineer

During the past two decades, I was part of the core engineering teams of the following startups and scale-ups:

- Prisma, engineering features of the query engine and the integration of Rust and JS runtimes.
- Hasura, scaling the backends of the cloud offering.
- Fastly, implementing and scaling real-time and historical analytics.
- GitHub, on all things backend and data-infrastructure.
- And a few others

Source https://mff.io/




# Part I. A fulfilling career

## What's a Career Path?

- The sequence of professional roles, experiences, and developmental milestones that an individual follows throughout their working life.
- Two types:
  - Linear: Progressive roles, in the same area of expertise, industry.
  - Non-linear. You might switch
    - Roles, Areas of expertise, Industries...
    - Engagement models
    - **Even professions altogether**

## Your career will evolve

### Take ownership early

- You will be a different person in 5, 10, 20, 30 years
- What's hot today, might be dead tomorrow - adaptability is key
- Your early career decisions shape future opportunities
- **It's not about having all the answers, but making an effort of shaping your intention.**

# Building a fulfilling career

## A step-by-step guide

1. Knowing oneself
2. Balancing short and long terms
3. Making a plan and sticking to it for while
4. Review and adjust

# The first step: knowing oneself

## Where in these axes do you see yourself today?

**1. Stability vs. Risk and Reward**

- **Stability**: Developer at a government agency or a large bank (big corp).
- **Risk and Reward**: Cofounder or CTO at a tech startup.

**2. Technical Specialization vs. Generalism**

- **Technical Specialization**: Machine Learning Engineer at a deep tech scaleup.
- **Generalism**: Full-stack developer at a fast-growing startup.

**3. Autonomy vs. Structure**

- **Autonomy**: Solopreneur / Freelance developer
- **Structure**: Developer at a big corp like Microsoft

[ See appendix 1 ]

# The second step: balancing short & long terms

## Struggle now… or later

- [A lot of] Short-term pleasure often leads to [lots of] long-term difficulties
  - e.g., procrastination, lack of resilience, bad habits
- [A moderate] Short-term struggle often leads to [lots of] long-term rewards
  - e.g., discipline, mastery, success.
- **Struggle tends to be unavoidable**, It's smart to take it early and grow out of it

# The second step: balancing short & long terms

## An example

- Alex's thinking on applying to a local consulting company instead of Netflix
- They don't feel as they can do it
- The reality is they can
- And the reality is it might be hard, because there are gaps to fill
- And the reality also is, that it might not be as hard as they think

https://www.ted.com/talks/carol_dweck_the_power_of_believing_that_you_can_improve

# The second step: balancing short & long terms

## Chose comfort

- They landed a job at the local consultancy firm right after graduating
- They've been there for 5 years, they're ok but not too motivated. They are 28 now.
- Her company realized their job can be done by a cheaper worker using an AI assistant. And the company is not betting on their growth.
- They would like to readjust and take some time off, but they have a mortgage to pay, and not many savings, what's worse, they were not exposed to many different and challenging problems. So they are disadvantaged in a very competitive market.
- They are **struggling** now.

## Chose struggle

- They enrolled in English lessons, found a mentor to guide them after sending an email to Miguel, and started preparing for interviews. Lot's of work there. They had a part-time job at the same time, and had to miss a lot of plans with friends. They are **struggling** now.
- They interviewed and got rejected. They reflected on why
- They decided to keep preparing
- They decided to apply to another very demanding company.
- They landed the job.
- They worked hard for 4 years, and learnt a lot. They are 28 now.
- They paid their first house
- They are motivated by the problems they're working on, they've got financial comfort and they are planning on taking some time for them before switching jobs to an interesting company that reached out to them.

https://www.ted.com/talks/carol_dweck_the_power_of_believing_that_you_can_improve

# The second step: balancing short & long terms

## The morale

- **The previous example was fictional, and reductionist,** but it's definitely plausible, specially as we move towards AGI and less competent roles could be replaceable.
- Some deliberate ~~struggles~~ **bets** yield long-term comforts.
  - 🏗️ Build strong fundamentals
  - 🔬 Learn to learn new things effectively
  - 🔥 Choose hard problems, even if you can fail at them.
  - 🤲 Find mentors and help others grow

https://www.ted.com/talks/carol_dweck_the_power_of_believing_that_you_can_improve

# The third step: making a plan

## And sticking to it for a while

- Plan an execute:
  - What are my gaps
  - What things should I do to fill them? How do I ensure I filled them?
  - What's the time horizon?
  - Find the critical path
  - Small but steady: low hanging fruits first and quick wins often

# The last step: Review and Adjust

- New information will give you insights:
  - You will discover new learning tactics that work better for you
  - Ineffective approaches can be replaced with more effective ones
- Often, **what you thought you wanted is not what you actually want**
  - It's normal to realise a path isn't the right fit
  - Let it sit for a while, but if the feeling persists, adjust your plan
- **Stay focused on progress, but be flexible enough to adapt**

# Roles, Companies, Compensation and More.

## Roles
### Individual Contributors

- Technical Focus
- No Direct Reports
- Career Growth Without Management
- Deep Expertise
- Mentorship & Influence
- Success based on impact of contributions

## Roles
### Engineer Managers

- Not necessary software engineer
- Team Leadership & People Management
- (Good ones) Project & Execution Management
- (Good ones) Technical oversight and decision making
- Cross-team collaboration
- **Lots of politics**
- In scale-ups and big companies, they can grow into VPoE, or C-level roles.

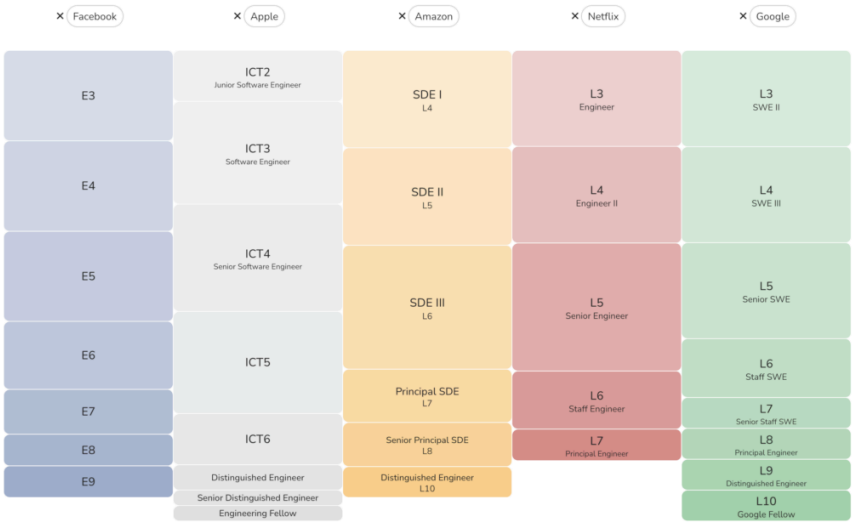
**In any hierarchical organisation, the most contributing factor to your growth is how well you help your manager do their job.**

However, growth might not equate to success.

# Roles

## Staff+ Engineers and Technical Leaders

- They are ICs, and while not managing people directly, they exercise leadership
  - Team efforts prioritisation
  - Cross-team collaboration
  - Identification of business opportunities and problems
  - Outreach to third parties (from other companies) to address strategic technical alliances
  - Interviewing, and hiring other engineers




# Roles

## VPs and C-level execs

- Decision makers
- Progression from either:
  - Being a founder
  - Being a manager
- Strategic focus:
  - Building the tech culture
  - Partnership and agreements
  - Less technical work

# Interlude: A bit of Jargon (and then a Pause)

- **VC (Venture Capital):** Investment in private companies in exchange for equity, aiming for high returns.
- **VC-backed company:** Receives VC funding, leading to ownership dilution and reduced founder control.
- **Series (Seed, A, B, C):** Successive funding rounds where companies trade equity for capital to scale.
- **Bootstrapping:** Growing a company with minimal or no VC funding to retain full ownership and control.
- **Stock options:** An incentive to employees, by which the company gives them the option to buy shares of the company while it's still private.
- **IPO:** The process by which a company becomes public, and allows people to trade shares of the company.
- **Product-market fit:** a stage of a company in which they are validating that there is enough market demand for their product to build a sustainable business out of it.

# Companies
## Big Tech (FAANG & Similar)

**Description:** Large, well-established companies with global reach and structured career paths.
**Examples:** Google, Meta, Apple, Microsoft, Amazon.

✅ High salary, benefits, and job security.
✅ Great mentorship, best practices, and specialization.
✅ Strong résumé boost for future jobs.

⚠️ Most require re-location.
⚠️ Bureaucracy, slow decision-making.
⚠️ Less ownership over projects for non-senior / staff positions.

🟢 **Best for:** Engineers seeking stability, prestige, and deep specialisation.

# Companies
## Mid-size Tech Companies

**Description:** Established, tech companies with structured but flexible environments.
**Examples:** Cloudflare, Datadog, GitHub.

✅ Strong engineering culture.
✅ More ownership than FAANG, but still structured.
✅ Good pay, work-life balance, and growth opportunities.
✅ More opportunities for remote work than FAANG (GitHub, Stripe, Datadog, Grafana…)

⚠️ Some bureaucracy, but less than FAANG.
⚠️ Not as prestigious but still well-respected.

🟢 **Best for:** Engineers wanting a balance of structure, impact, and growth, still acquiring some reputation.

# Companies
## Scale-ups

**Description:** High-growth startups with VC funding, scaling teams, and aggressive expansion.
**Examples:** Notion, Figma, OpenAI, Deel, Brex.

✅ Rapid career growth, exposure to big challenges.
✅ Equity upside if the company succeeds.
✅ Modern tech stack, fewer legacy systems.

⚠️ High workload, sometimes chaotic processes.
⚠️ Risk of layoffs or business failure.

🟢 **Best for:** Engineers who want fast learning, risk-taking, and high rewards.

# Companies
## Start-ups

**Description:** Small, young startups still finding product-market fit. High risk, high learning.
**Examples:** YC startups, stealth mode startups.

✅ Extreme learning, engineers wear many hats.
✅ High ownership, work closely with founders.
✅ Potential big equity rewards if successful.

⚠️ Unstable pay, risk of failure, unclear direction.
⚠️ Long hours, unpredictable work environment.

🟢 **Best for:** Engineers who thrive in chaos and want hands-on startup experience, or seek to create their own businesses soon and want to familiarise with the inners of creating a tech startup.

# Companies
## Bootstrapped businesses

**Description:** Self-funded, profitable companies focused on sustainability over rapid growth.
**Examples:** 37signals, Gumroad, TechCrunch.

✅ High autonomy, small but meaningful teams.
✅ No VC pressure—focus on product, not hypergrowth.
✅ Generally better work-life balance than VC-backed startups.

⚠️ **Generally** slower salary growth, fewer resources
⚠️ Can be niche, less known in the industry.

🟢 **Best for:** Engineers who value autonomy, product focus, staying for longer, and craftsmanship.

# Companies
## Boutique Agencies / Software Consultancies

**Description:** Small firms that build custom software for clients, often high-quality codebases.
**Examples:** ThoughtBot, ThroughtWorks, Hashrocket, 8th Light.

✅ Work on diverse projects, fast skill development.
✅ Strong technical culture, less bureaucracy.

⚠️ Pay may be lower than product companies.
⚠️ Work depends on client demands, less creative freedom.

🟢 **Best for:** Engineers who enjoy variety, consulting, and technical craftsmanship.
💥 **Software factories are excluded**

# Companies
## Public Sector

**Description:** Tech roles in government or public institutions, often stable but super slow-moving.
**Examples:** US Digital Service, UK Gov Digital, e-Estonia.

✅ Strong job security, good work-life balance.
✅ Meaningful impact—technology for public good.
✅ Inclusive, diverse work environment.

⚠️ Bureaucracy, slow decision-making.
⚠️ Frequently: Outdated tech stacks, practises.
⚠️ Lower salaries
⚠️ Spanish national and regional agencies are well-known for their lack of focus on UX, craftsmanship, and solid operational practises.

🟢 **Best for:** Engineers seeking stability, predictable, work.

# Companies
## Research centers / Universities

**Description:** Academic or corporate research labs focused on cutting-edge innovation and fundamental research.

✅ Work on advanced tech (AI, quantum computing, robotics, etc.).
✅ Scientific and academic contribution.
✅ Less pressure for monetisation compared to industry.

⚠️ Underfunded.
⚠️ Salaries lower than industry roles.
⚠️ Work may be theoretical, slow to impact real-world applications.

🟢 **Best for:** Engineers interested in deep research, academia, or foundational tech breakthroughs, or with a vocation for research and teaching.

# Companies
## Entrepreneurship / Build Your Own Business

**Description:** Starting and growing your own company, from indie projects to VC-backed startups.
**Examples:** Indie hackers, startup founders, bootstrapped SaaS, YC startup founders.

✅ Ultimate ownership—build something from scratch.
✅ High potential upside if the company succeeds.
✅ Learn everything: tech, business, marketing, hiring.

⚠️ High risk, most startups fail.
⚠️ Unstable income, high stress, long hours.

🟢 **Best for:** Engineers who want control, are resilient and intrinsic motivated, are generalists, and are willing to take risks.

# Companies
## Summary

| Company Type | Autonomy | Compensation | Risk | Generalist/Specialist | Intensity |
|---|---|---|---|---|---|
| **FAANG / Big Tech** | 3 | High | Medium | Mostly Specialist | Medium |
| **Mid-sized Tech** | 5 | High | Medium | Balanced | Medium |
| **Scale-Ups** | 7 | Highest (talent bait) | Medium | Balanced | High |
| **Early-Stage Startups** | 9 | Medium | High | Mostly Generalist | Very High |
| **Agencies** | 2 | Low | Low | Balanced | Medium |
| **Government** | 1 | Low | Lowest | Mostly Generalist | Low |
| **Research Labs / Universities** | 8 | Medium/Low | Low | Mostly Specialist | Medium |
| **Entrepreneurship / Startups** | 10 | Low (Highest potential) | Highest | Mostly Generalist | Very High |

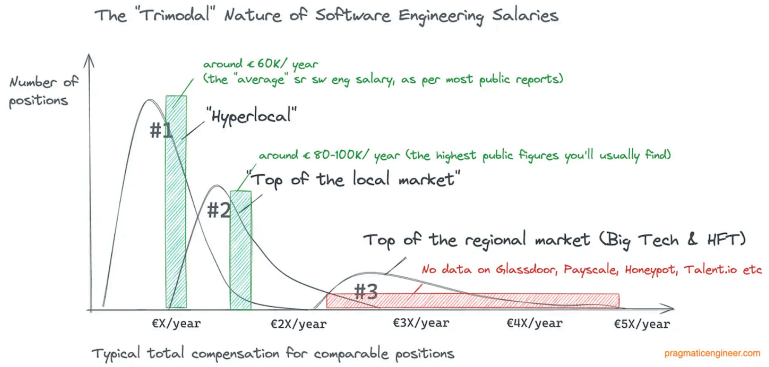
Disclaimer: All metrics can have a wide variance for specific instances of each company type. Data reflects the author's perspective over a typical case.

# Companies
## Author's opinion

| Company Type | Autonomy | Compensation | Risk | Generalist/Specialist | Intensity |
|---|---|---|---|---|---|
| **FAANG / Big Tech** | 3 | High | Medium | Mostly Specialist | Medium |
| **Mid-sized Tech** | **5** | **High** | **Medium** | **Balanced** | **Medium** |
| **Scale-Ups** | **7** | **Highest (talent bait)** | **Medium** | **Balanced** | **High** |
| **Early-Stage Startups** | 9 | Medium | High | Mostly Generalist | Very High |
| **Agencies** | 2 | Low | Low | Balanced | Medium |
| **Government** | 1 | Low | Lowest | Mostly Generalist | Low |
| **Research Labs / Universities** | 8 | Medium/Low | Low | Mostly Specialist | Medium |
| **Entrepreneurship / Startups** | 10 | Low (Highest potential) | Highest | Mostly Generalist | Very High |

Disclaimer: All metrics can have a wide variance for specific instances of each company type. Data reflects the author's perspective over a typical case.

# On compensation



# Time to get real

## Your career navigation map

### Turning uncertainty into strategy

- Yourself (5')
- The moonshot (5')
- The gaps to fill (7')
- A plan for the next 12 months (10')
- Your personal commitment

## Yourself (5')

- Imagine yourself after graduation
- What kind of work are you doing?
- Place yourself in the different axes provided

## Appendix: Career axes

**1. Stability vs. Risk and Reward**

- **Stability**: Developer at a government agency or a large bank (big corp).
- **Risk and Reward**: Cofounder or CTO at a tech startup.

**2. Technical Specialization vs. Generalism**

- **Technical Specialization**: Machine Learning Engineer at a deep tech scaleup.
- **Generalism**: Full-stack developer at a fast-growing startup.

**3. Autonomy vs. Structure**

- **Autonomy**: Solopreneur / Freelance developer
- **Structure**: Developer at a big corp like IBM or Accenture.

**4. Social Impact vs. Economic Impact**

- **Social Impact**: Software engineer at a nonprofit or healthcare startup.
- **Economic Impact**: Algorithmic trading engineer at a financial big corp like Goldman Sachs.

**5. Work-Life Balance vs. High Commitment**

- **Work-Life Balance**: Developer at a remote-first scaleup or a big corp like GitLab.
- **High Commitment**: Engineer at a hyper-growth startup.

**6. Individual Work vs. Team Collaboration**

- **Individual Work**: Freelance developer working on independent projects.
- **Team Collaboration**: Engineer at Google or a startup building a new product.

**7. Innovation vs. Maintenance**

- **Innovation**: Researcher in advanced AI at a university or engineer at a deep tech startup.
- **Maintenance**: Legacy system maintainer at a telecom company or bank.

**8. Initial Salary vs. Growth Potential**

- **Initial Salary**: Consultant at Deloitte or developer at a FAANG company.
- **Growth Potential**: Early employee at a startup with equity stakes.

## The moonshot (5')

- What kind of role, in which kind of company would you be doing if you had all the skills necessary? Examples:
  - SWE at NASA designing and implementing the new generation of their flight-control engine.
  - CEO of a startup building augmented reality systems for industrial simulation.
  - Researcher and Professor at the University of Oviedo working on the formal validation of computer-generated programs
  - Lead Engineer of the most popular Open Source ML inference runtime
  - ...

## Filling the gaps (5')

- Think
  - Skills and Knowledge - What should I learn?
  - Networking and Mentoring - Who can guide me?
  - Experimentation - How can I test my options?

## A plan for the next 12 months (10')

- Find 3 things you can do over the next year that can move you closer to your moonshot.
  - How can you fill-in your gaps more effectively?
  - What are some things your could learn that are not in the Study Plan?
  - Who can guide you further? Who can you team-up with?
  - What initiatives can you enrol-in?
  - What companies can you engage with?

## Commit to it (5')

**In the next year, I will do [ACTIONS], so I will improve [GAPS] and eventually be better prepared to work as [MOONSHOT].**

# Anyone sharing?

# Ask Me Anything

## Interesting reads

- On learning to learn new things:
  - Pragmatic Thinking and Learning - Refactor Your Wetware by Andy Hunt.
  - Meta Learning - How To Learn Deep Learning and Thrive in The Digital World by Radek Osmulski
- Early career choices influence future outcomes:
  - “The Matthew Effect in Science” by Robert K. Merton (1968)
- On making plans and sticking to them, it’s beneficial to understand how long-lasting habits are formed:
  - “Atomic Habits” by James Clear

## Interesting reads

- On roles:
  - The Pragmatic Programmer: Your Journey To Mastery, 20th Anniversary Edition by David Thomas and Andrew Hunt
  - Staff Engineer: Leadership beyond the management track by Will Larson
  - The Manager's Path: A Guide for Tech Leaders Navigating Growth and Change by Camille Fournier
- General career guidance
  - newsletter.pragmaticengineer.com - General knowledge about tech and startups highly relevant for software engineers by Gergely Orosz
  - Best articles:
    - https://blog.pragmaticengineer.com/advice-to-myself-when-starting-as-a-software-developer/
    - https://blog.pragmaticengineer.com/the-developer-culture-test/
    - https://blog.pragmaticengineer.com/software-engineering-salaries-in-the-netherlands-and-europe/

# Resources to prepare for a job at a FAANG

- https://www.crackingthecodinginterview.com/ - The most iconic book on code interview preparation
- https://medium.com/@sentalkssane/a-beginners-guide-to-system-design-76d64689788b - insights on how to prepare for system design interviews at FAANG and other tech companies.
- https://www.levels.fyi/ - get insights on compensation and salaries of many companies
- https://www.kalzumeus.com/2012/01/23/salary-negotiation/ - salary negotiation by Patrick McKenzie

# Appendix: Career axes

**1. Stability vs. Risk and Reward**

- **Stability**: Developer at a government agency or a large bank (big corp).
- **Risk and Reward**: Cofounder or CTO at a tech startup.

**2. Technical Specialization vs. Generalism**

- **Technical Specialization**: Machine Learning Engineer at a deep tech scaleup.
- **Generalism**: Full-stack developer at a fast-growing startup.

**3. Autonomy vs. Structure**

- **Autonomy**: Solopreneur / Freelance developer
- **Structure**: Developer at a big corp like IBM or Accenture.

**4. Social Impact vs. Economic Impact**

- **Social Impact**: Software engineer at a nonprofit or healthcare startup.
- **Economic Impact**: Algorithmic trading engineer at a financial big corp like Goldman Sachs.

**5. Work-Life Balance vs. High Commitment**

- **Work-Life Balance**: Developer at a remote-first scaleup or a big corp like GitLab.
- **High Commitment**: Engineer at a hyper-growth startup.

**6. Individual Work vs. Team Collaboration**

- **Individual Work**: Freelance developer working on independent projects.
- **Team Collaboration**: Engineer at Google or a startup building a new product.

**7. Innovation vs. Maintenance**

- **Innovation**: Researcher in advanced AI at a university or engineer at a deep tech startup.
- **Maintenance**: Legacy system maintainer at a telecom company or bank.

**8. Initial Salary vs. Growth Potential**

- **Initial Salary**: Consultant at Deloitte or developer at a FAANG company.
- **Growth Potential**: Early employee at a startup with equity stakes.

# Stay in touch


- miguel@mff.io
- https://thesignal.team
- miguelff @ { linkedIn, GitHub, bsky.social }
- mff.io


# Thank you!