

{quality development}

How to properly write your project



Università degli Studi di Catania
Dipartimento di Matematica e Informatica

2023





\$ whoami

Hi, I am Stefano Borzì 

Open-source developer

Programmer Analyst Senior at FedEx

PhD student at University of Catania



Course introduction

- Shell UNIX
- Git
- GitHub
- Opensource
- Python
- Unit-test
- SOLID principles
- CI / CD pipeline & tools

Spotted DMI

Se si erogasse un nuovo corso al DMI, lo vorreste inerente a

Anonymous Poll

34% Come scrivere codice di qualità



36% Utilizzare professionalmente git e GitHub



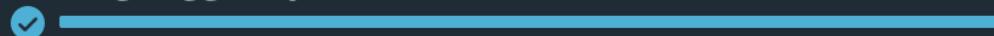
15% Come usare il terminale su Linux



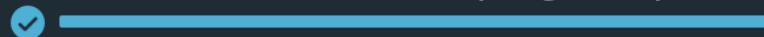
17% Unit-test, end-to-end test



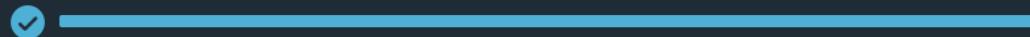
31% Linguaggio Python



23% Come funzionano le comunità open-source e come contribuire a eventuali progetti open



32% Approfondire web programming mediante lo studio di framework moderni



176 votes

436 10:28



Role

As a senior software developer, you will be responsible for designing solutions to problems from the group up, reviewing code from members of your team, TDD all of your code, help teammates improve in the technical skills and knowledge, whilst also working closely with product managers and company wide stakeholders. You will be part of an enthusiastic team responsible for product development of platform services, which is mostly high scale backed services. You will have the ability to choose the right tools and methods that you decide are appropriate to solve any problems, but will be supported by experienced team members.

Senior Software Developer Requirements:

- 3 years experience in Javascript and/or Typescript
- Experience with React framework
- Proficiency with Git source control
- Efficient HTML and CSS ability
- Good communication skills in written, verbal, and code

Offer:

- Up to £90,000

WHAT WE'D LOVE YOU TO DO (AND LOVE DOING)

- Develop high-quality and responsive user interfaces for web applications using modern frontend frameworks such as React, Angular.
- Collaborate with designers to create visually appealing interfaces that meet user needs and brand guidelines
- Work closely with product managers and backend engineers to integrate frontend components with backend APIs and databases
- Ensure the web applications are optimized for speed and performance, including minimizing load times and maximizing interactivity
- Write clean, modular, and maintainable code that can be easily understood and modified by other team members
- Participate in code reviews and provide constructive feedback to other team members
- Stay up-to-date with the latest frontend technologies and trends and bring new ideas and solutions to the team
- Ensure the web applications are accessible to all users, including those with disabilities
- Test and debug frontend components to identify and fix issues that arise during development
- Document technical specifications and provide support to other team members as needed

WE'D LOVE TO MEET YOU IF

Must Haves-

- Strong proficiency in JavaScript, TypeScript, HTML, CSS, and Node.js
- Experience with modern frontend frameworks such as React
- Strong problem-solving skills and attention to detail
- Excellent communication and collaboration skills
- Experience with version control tools such as Git
- Commitment to continuous learning and self-improvement

- Manage, lead and coordinate projects
- Design, develop, and maintain our platform
- Create high-quality software that scales
- Help our analysts work more efficiently
- Develop new features and systems
- Take ownership of parts of our platform

Who You Are

- You have a degree in Computer Science or equivalent, or relevant experience
- You have outstanding Python skills
- ...with at least 5 years of professional experience
- You have experience with Docker and Kubernetes
- You have a good understanding of NoSQL databases like MongoDB and RabbitMQ
- You feel comfortable building CI/CD pipelines
- You know your way around software documentation, and continuous integration tools like Jenkins and CircleCI
- ...and you don't mind collecting some Kubernetes skills

- **minimo di 4 anni esperienza con PHP e MySQL;**
- **minimo 3 anni di esperienza con Laravel;**
- **minimo 2 anni di esperienza con Vue.js;**

Senior Software Engineer, Data Infrastructure

Vinted · Amsterdam, North Holland, Netherlands (Holland)

Apply ↗

Save

...

In this position, you'll

- Work within a team and participate in decision-making, planning and implementing data infrastructure solutions that improve our users' experience
- Develop scalable, distributed, cost-effective solutions for data loading, processing, transformation and analysis
- **Collaborate with other developers through GitHub pull requests and code reviews**
- Write and maintain IaC (Terraform, Terragrunt)
- Implement and adopt DevOps practices in our platform
- Monitor and troubleshoot platform issues. We monitor our code and infrastructure through Kibana, Grafana and other tools
- Consult DSA and other engineering teams on best practices and implementation
- Apply Agile and Lean principles in your work
- Write unit tests for the code you write

Our Tech Stack

- **Backend Code:** Rust, Tokio, Cargo, Git (<https://github.com/cowprotocol/services>)
- **Infrastructure:** Pulumi, AWS, Docker, Kubernetes, Grafana, Prometheus, Elasticsearch
- **Other codebases:** Typescript, Python, Solidity

Who you are

- You have 3+ years of experience with system programming, preferably in Rust (C++ or similar)
- You have excellent computer science, programming, and algorithmic skills
- You care deeply about the quality, readability, observability, and testability of your code
- You proactively give technical direction to improve performance, prevent issues, optimise data us
- You are a team player, a strong communicator, and gladly share your expertise with others
- You collaborate effectively with a remote-first team on a large, open-source codebase
- Excited to explore and gain experience with Ethereum

Your skills

To begin with, we are looking for junior **Data Scientists** with the following essential skills and experiences:

- Master of Science degree in Mathematics, Statistics, Computer Science, Engineering Physics, Electrical Engineering or related fields.
- Programming skills with one or more of the following languages: Python, R, Scala.
- Theoretical background in Machine Learning, Statistical Data Analysis, and/or Optimization.
- Great communication skills and good at explaining complex technical concepts to business leaders in a compelling way.

It is meritorious if you also have some of the following experiences or skills:

- Hands-on experience of implementing analytics & machine learning projects.
- Hands-on experience of querying data from databases and big data platforms.
- Data visualization with Tableau, Power BI or other business intelligence tools.
- DevOps for data & analytics: Git, MLflow, Docker, Kubernetes, and CI/CD tools (Jenkins, GitHub Actions)
- Experience of working with Cloud platforms, for example MS Azure, GCP, AWS, and Databricks.

I graduated in Computer Science in the early 2000s.

When I took a Databases class, NoSQL didn't exist.

When I took a Computer Graphics class, OpenGL didn't support shaders.

When I took a Computer Security class, no one knew about botnets yet.

When I took an Artificial Intelligence class, deep learning didn't exist.

When I took a Programming Languages class, reactive programming wasn't a "thing".

When I took a Distributed Systems class, there was no Big Data or cloud computing.

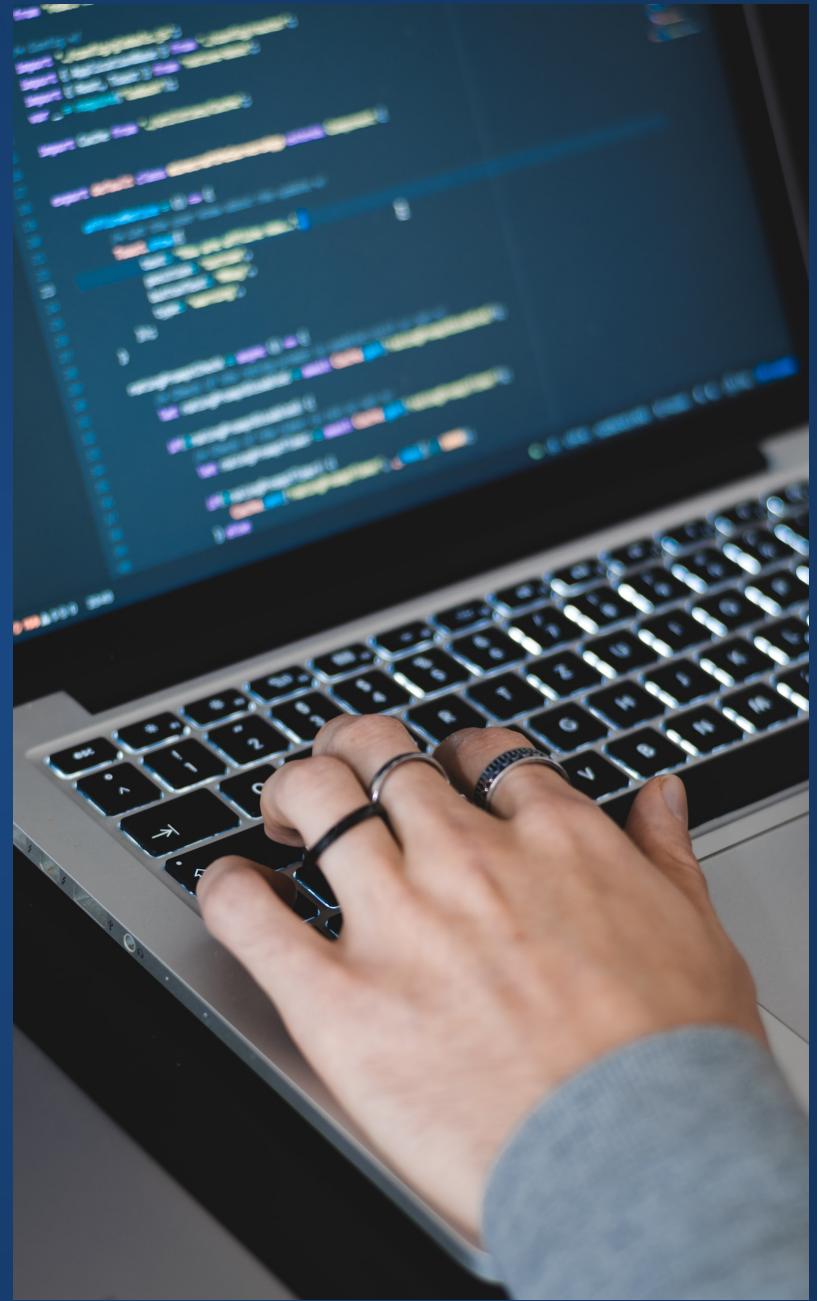
When I took an Operating Systems class, hypervisors didn't exist (in PCs at least).

When I took a Networking class, there was no wifi in my laptop or internet in my phone.

Learn the fundamentals. The rest will change anyway.|

Exam

Mini-project in a group of two

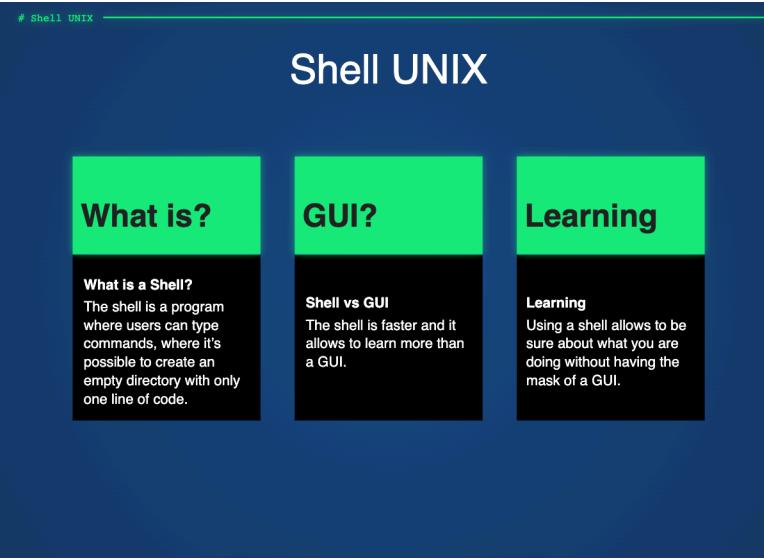




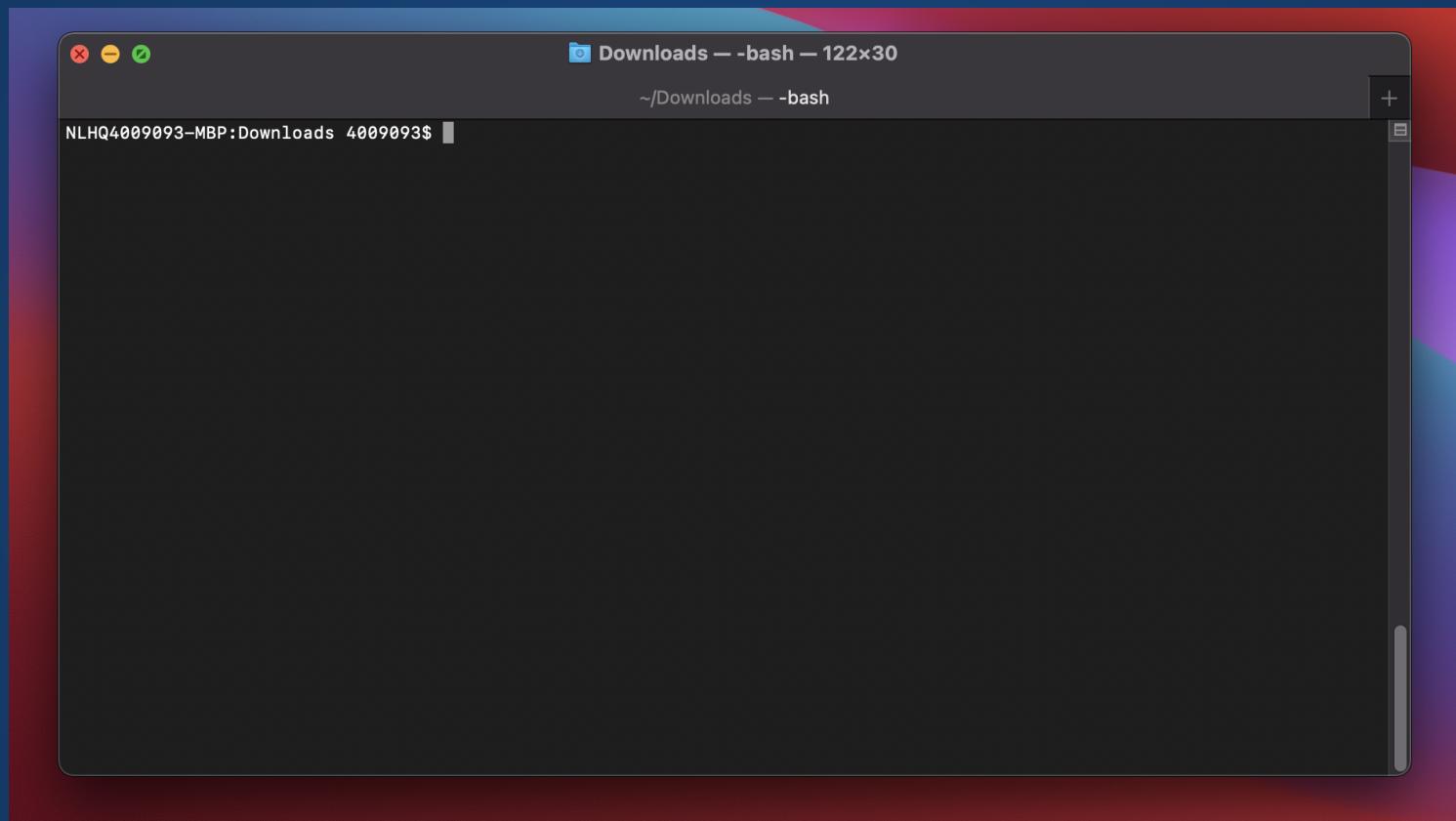
Shell UNIX

Mandatory to use for servers





Shell UNIX -



Shell UNIX

```
ubuntu@ubuntu-VirtualBox:~$ echo "Hello World"
Hello World
ubuntu@ubuntu-VirtualBox:~$
```

```
$ nano First.sh
```

```
#!/bin/bash
echo "Hello World"
```

GNU nano 2.8.6

File: First.sh

Modified

```
#!/bin/bash
echo "Hello World"
```

^G Get Help **^O** Write Out **^W** Where Is **^K** Cut Text **^J** Justify **^C** Cur Pos
^X Exit **^R** Read File **^V** Replace **^U** Uncut Text **^T** To Linter **^** Go To Line

```
$ bash First.sh
```

```
#!/bin/bash

# Add two numeric value
((sum=25+35))

#print the result
echo $sum
```

```
$ bash comment_example.sh
```

```
ubuntu@ubuntu-VirtualBox:~/code$ bash comment_example.sh
60
ubuntu@ubuntu-VirtualBox:~/code$
```

Useful commands #1

Command	Description
ls	show files and directories
man	read manual
cd	open directories
mv	move (or rename) file
cp	copy file or directory (-r)
rm	remove file or directory (-r)
pwd	(current) path working directory
mkdir	create a directory
nano, vim, ed	terminal text editor

cd

\$ cd directory_name

\$ cd ..

ls

\$ ls

\$ ls path

\$ ls -f

\$ ls test_*

\$ ls *.cpp

\$ ls ?.cpp

rm

\$ rm path/of/file

\$ rm -r path/of/directory/

\$ rm -rf path/of/directory



The shell does not have a trash bin:
once something is deleted, it's
really gone.

pwd

\$ pwd - return the absolute path of the working directory

/home/helias/ = ~/

Pathnames

- .. upper directory
- . current directory

absolute path

/home/helias/Desktop/file.png

relative path

home/helias/Desktop/file.png

./home/helias/Desktop/file.png

Desktop/file.png = ./Desktop/file.png

TAB button



superpower: **autocomplete**

Useful commands #2

Command	Description
cat	displays the contents of its inputs
head	displays the first 10 lines of its input
tail	displays the last 10 lines of its input.
command > [file]	redirects a command's output to a file (overwriting any existing content).
command >> [file]	appends a command's output to a file.
[first] [second]	it is a pipeline: the output of the first command is used as the input to the second.
grep [filter]	grep part of input based on filter

Useful commands #3

Command	Description
Ctrl+C	halts the current command
Ctrl+Z	stops the current command, resume it in the background
Ctrl+R	type to bring up a recent command
Ctrl+L, clear	clean the terminal
!!	repeats the last command
exit	log out of current session
more, less	read a file part by part

Useful commands #4

Command	Description
\$@	take all inputs parameters
\$1	take the first input parameter
\$2	take the second input parameter
wget	"web get", download from URL
curl	transfer a URL, send http requests
zip	it allows to compress directories and/or files
mysql / mysqldump	client mysql

alias

\$ man

\$ What manual page do you want?

\$ woman

\$ -bash: woman: command not found

\$ alias woman = man

\$ woman

\$ What manual page do you want?

tmux - cheatsheet

\$ tmux new -s new-session

\$ tmux a -t new-session

\$ Ctr+B + D # detach session

Exercises #1

1. make a directory called **first**
2. change directory to the **first** folder
3. create a file called **person.txt**
4. change the name of **person.txt** to **another.txt**
5. make a copy of the **another.txt** file and call it **copy.txt**
6. remove the **copy.txt** file
7. make a copy of the **first** folder and call it **second**
8. delete the **second** folder
9. Create a bash script that executes automatically all these tasks
waiting **1** second for each task using **sleep**

Exercises #2

1. What does the `man` command do? Type in `man rm`. How do you scroll and get out?
2. Look at the `man` page for `ls`. What does the `-l` flag do? What does the `-a` flag do?
3. Type the following command to download and save the contents of google.com:

```
curl https://www.google.com > google.html
```

4. Use `less` to look at the contents of `google.html`.
5. Look at the `man` page for `less`. Read the section on `-p (/pattern)`. Search for the text `hplogo` in the `google.html` file.
6. What is the shortcut to `clear` the terminal?
7. What is an `absolute path`?
8. What is an `relative path`?
9. What do the `r` and `f` flags do with the `rm` command?

alias exercises

\$ zipthis DIRECTORY

\$ notify MESSAGE

\$ echo "hello world" | espeak -v it

Challenge: extend notify-me

Extend the bash script `notify.sh` into other bash scripts that allow to send single file or directory (zipped), using:

\$ tg-send path/of/the/file/or/directory

Bonus feature:

- recognize if the file is more than **50 MB**, in that case the Telegram API will block you, so it is better to warn the user about it
- use telegram client (not bot) API to overcome the 50 MB limit and get **2 GB** as limit.

References:

- <https://github.com/Helias/Notify-me>
- <https://github.com/azerothcore/telegram-automated-db-backup>