



Explore Careers. Learn Stuff. Make Connections.

Zrive Applied Data Science

Intro to the program

Guillermo Barquero

About me



Guillermo Barquero
gb@zriveapp.com

- **Jun'22 – Now:** Founder & CTO @ **Zrive (London)**
- **Nov'20 – Jun'23:** Lead Data Scientist @ **Bother (London)**
- **Apr'18 – Nov'20:** Data Science Manager @ **Tesco (London)**
- **Jan'16 – Apr'18:** Data Scientist @ **ArcelorMittal (Asturias)**
- **Jan'14 – Jun'16:** SVM research project @ **EPI Gijon (Asturias)**

Claustro de profesores



Guillermo Barquero

CTO @ Zrive

Anteriormente: Data Science Manager @ Tesco y
Lead Data Scientist @ Bother

[Ver LinkedIn](#)



Guillermo Santamaria

Research Data Scientist @ Meta

Anteriormente: Lead Data Scientist @ Farfetch y
Principal Data Scientist @ Cabify

[Ver LinkedIn](#)



Sergio Rozada

ML Engineer @ Meta

Anteriormente: Data Science Manager @ Seedtag y
Senior Data Scientist @ BBVA

[Ver LinkedIn](#)



Pablo Alonso

Senior Data Scientist @ Scopely

Anteriormente: Data Scientist @ ArcelorMittal

[Ver LinkedIn](#)



Course overview (part I)

- 1 **Introducción al programa**
- 2 **Exploratory Data Analysis**
- 3 **Fundamentals of statistical learning**
- 4 **Model fitting**
- 5 **Analyse, diagnose and improve a model**
- 6 **Systems design**
- 7 **Business translation**

Course overview (part II)

Trabajo de 5 semanas en un proyecto real.

Trabajareis en proyectos reales propuestos por nuestras empresas colaboradoras.

1 Enfrentar un problema real de principio a fin.



Trabajareis en problemas reales propuestos por empresas donde tendreis que aplicar los cuatro pilares de la generación de valor:

- i. Traducir el problema de negocio a un problema de datos
- ii. Diseñar una solución algorítmica básica para solucionar
- iii. Implementar la solución en un MVP
- iv. Comunicar de forma efectiva vuestra solución

2 Trabajo en equipo



Trabajareis en grupos de 3-4 alumnos con el soporte de un mentor senior de Zrive y contaréis con miembros de la empresa como stakeholders de negocio. El objetivo es simular una situación real al máximo posible.

Funcionamiento del programa

1 Organización de la semana

- **Clase del martes a las 20.00 (se graba):**
 - Se presentan los conceptos a trabajar durante la semana. El profesor puede explicar con cierto grado de detalle algunos conceptos clave, pero se espera que los miréis en detalle por vuestra cuenta.
 - Se explica el problema práctico sobre el que trabajaréis durante el resto de la semana.
- **Trabajo autónomo:**
 - Se espera que estudiéis los conceptos propuestos para la semana haciendo uso de los materiales de preparación provistos.
 - Trabajaréis de forma autónoma los problemas prácticos para tenerlos solucionados antes de la clase del domingo.
- **Clase del domingo a las 20.00 (se graba):**
 - Repasaremos las soluciones a las que habéis llegado y el profesor mostrará su solución y explicará paso a paso el porqué de las decisiones que ha ido tomando para solucionarlo.

2 Code reviews

Todas las soluciones a los problemas propuestos tienen que subirse como una Pull Request semanal, que será revisada con comentarios por nuestros profesores para asegurar la incorporación de buenas prácticas de programación.

Funcionamiento del programa

3 1 to 1 sessions

- **Mock interviews:**
 - Idealmente en el finde presencial en Madrid (11 de Noviembre).
 - Para quien no pueda asistir se podrá hacer online durante esa misma semana (mitad del curso)
- **1 to 1 session:**
 - Tendréis la opción de bookear una sesión 1 to 1 online con alguno de los profesores donde podréis discutir: opciones profesionales o de carrera, cuestiones técnicas, hacer una nueva mock interview...

4 Finde presencial (11 de Noviembre)

- Sabado 11 (mañana): Haremos el workshop de bussiness translation en formato presencial.
- Sabado 11 (tarde): Mock interviews.
- Viernes 10: Visitas a empresas (TBD, solo si realmente tiene sentido y aporta valor).

5 Proyecto real

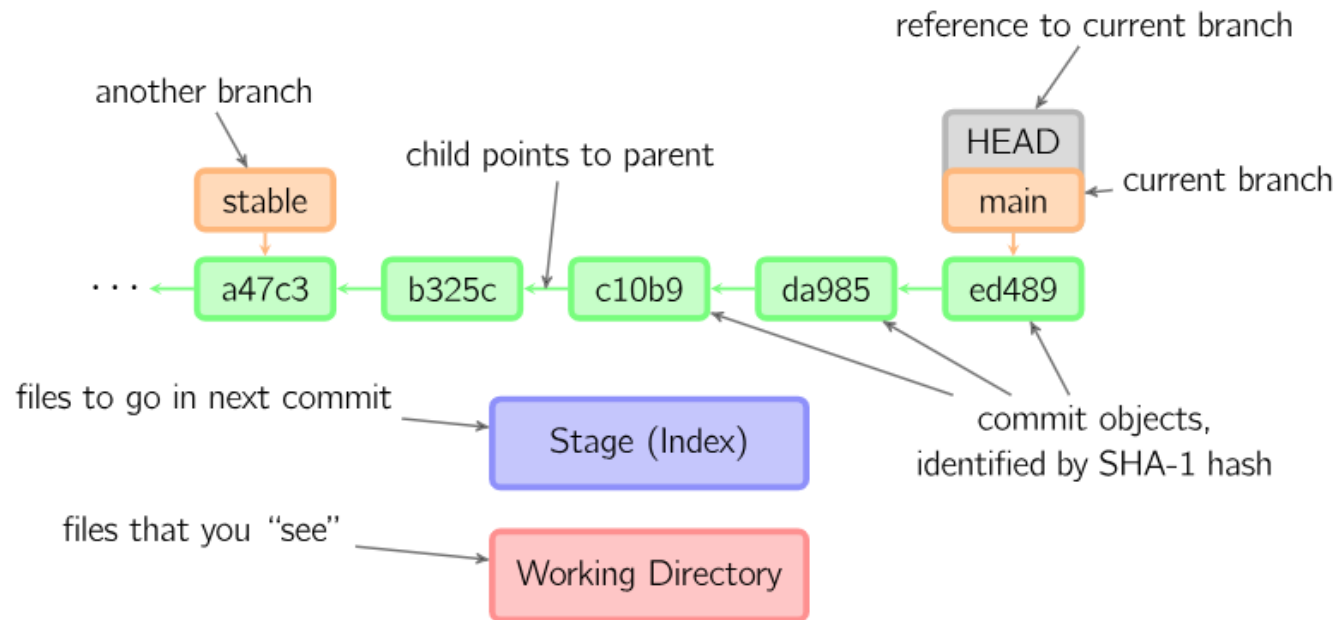
- Trabajo en equipo de 10-12h/semana.
- 2h/semana con el tech lead de vuestro proyecto + 1h a la semana (aprox) con stakeholders de la empresa.

DEV environment + REST API

Preparing a DEV environment

1 Github

1. Why version control is important?
2. Quick review of important bits: <https://marklodato.github.io/visual-git-guide/index-en.html>

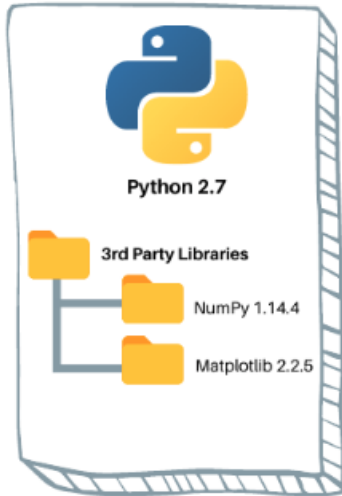


Preparing a DEV environment

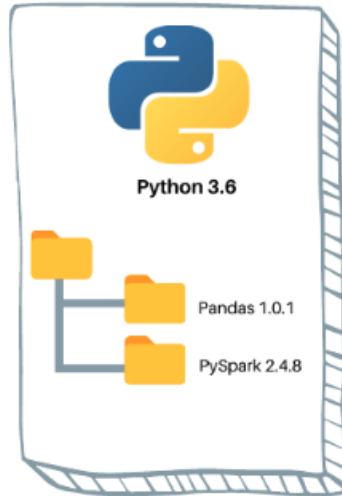
2 Python, pyenv, poetry

1. Why to use virtual environments?
2. Dependencies management

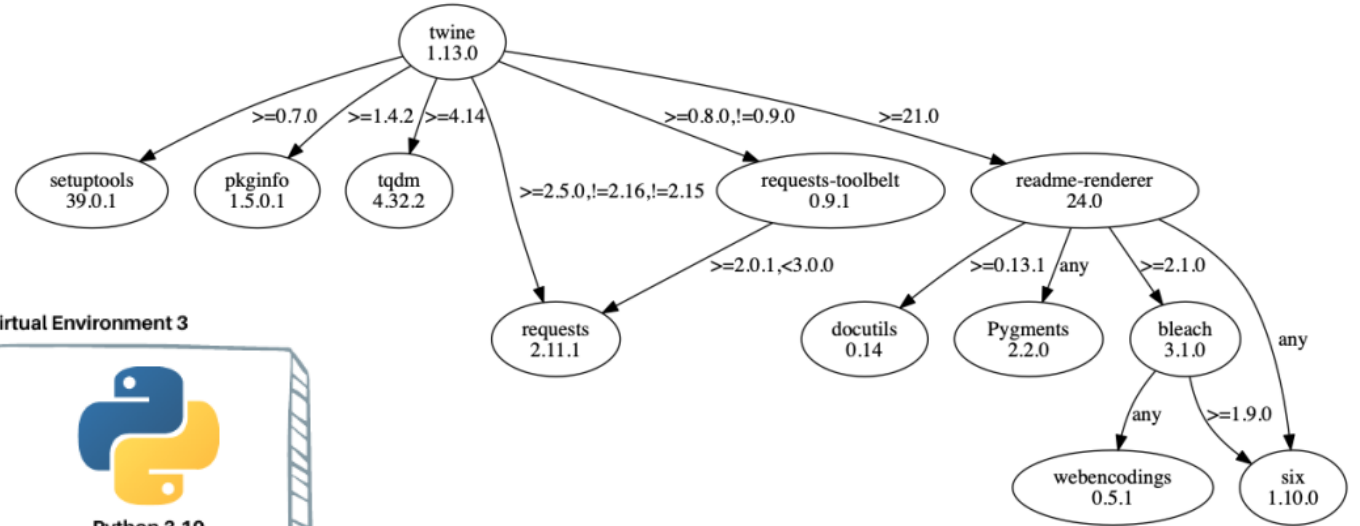
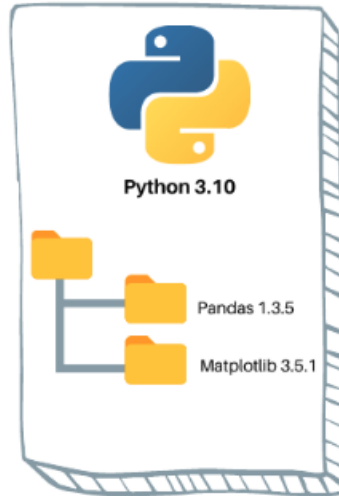
Virtual Environment 1



Virtual Environment 2

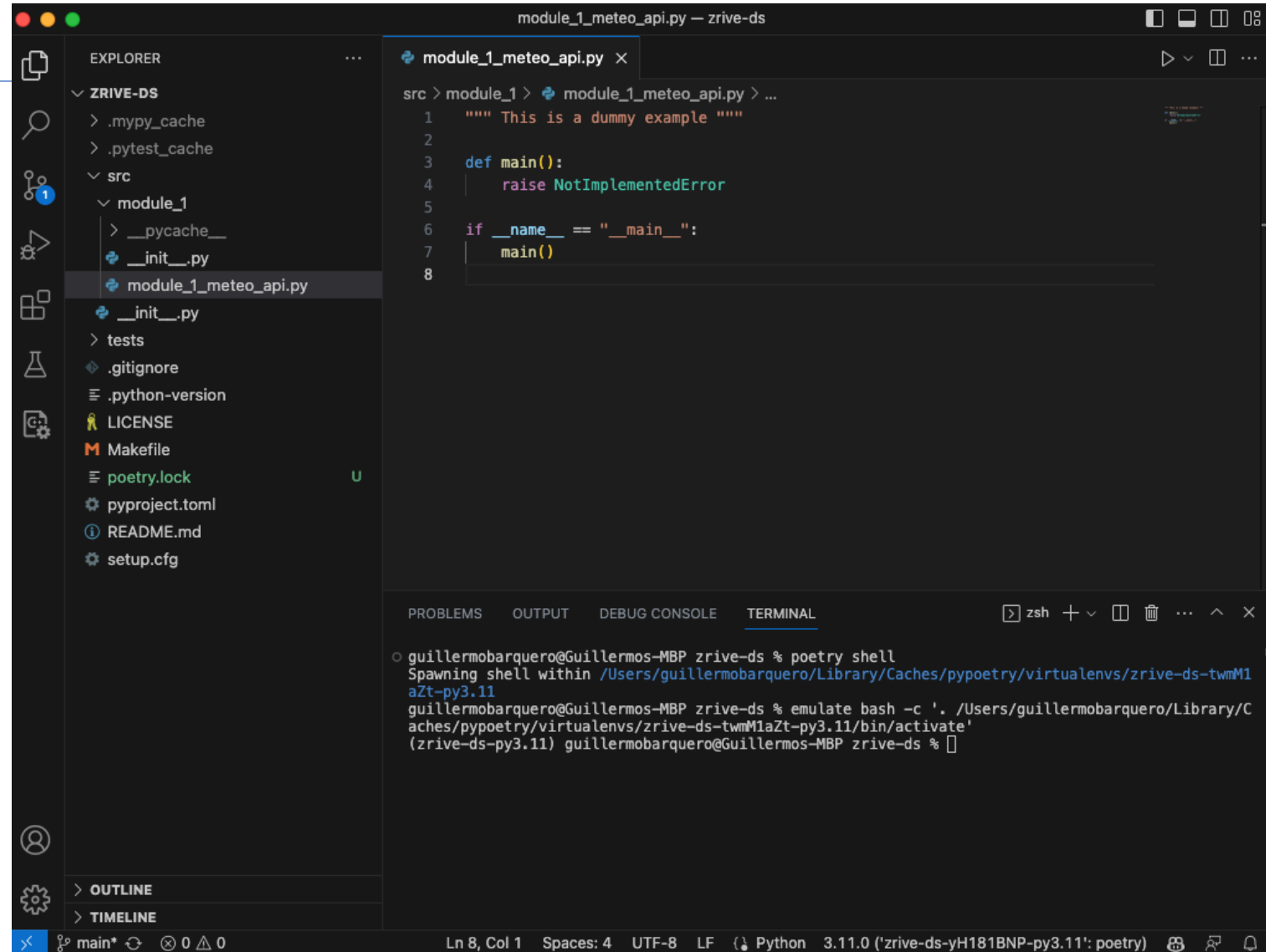


Virtual Environment 3



Preparing a DEV environment

3 IDE VScode



Preparing a DEV environment

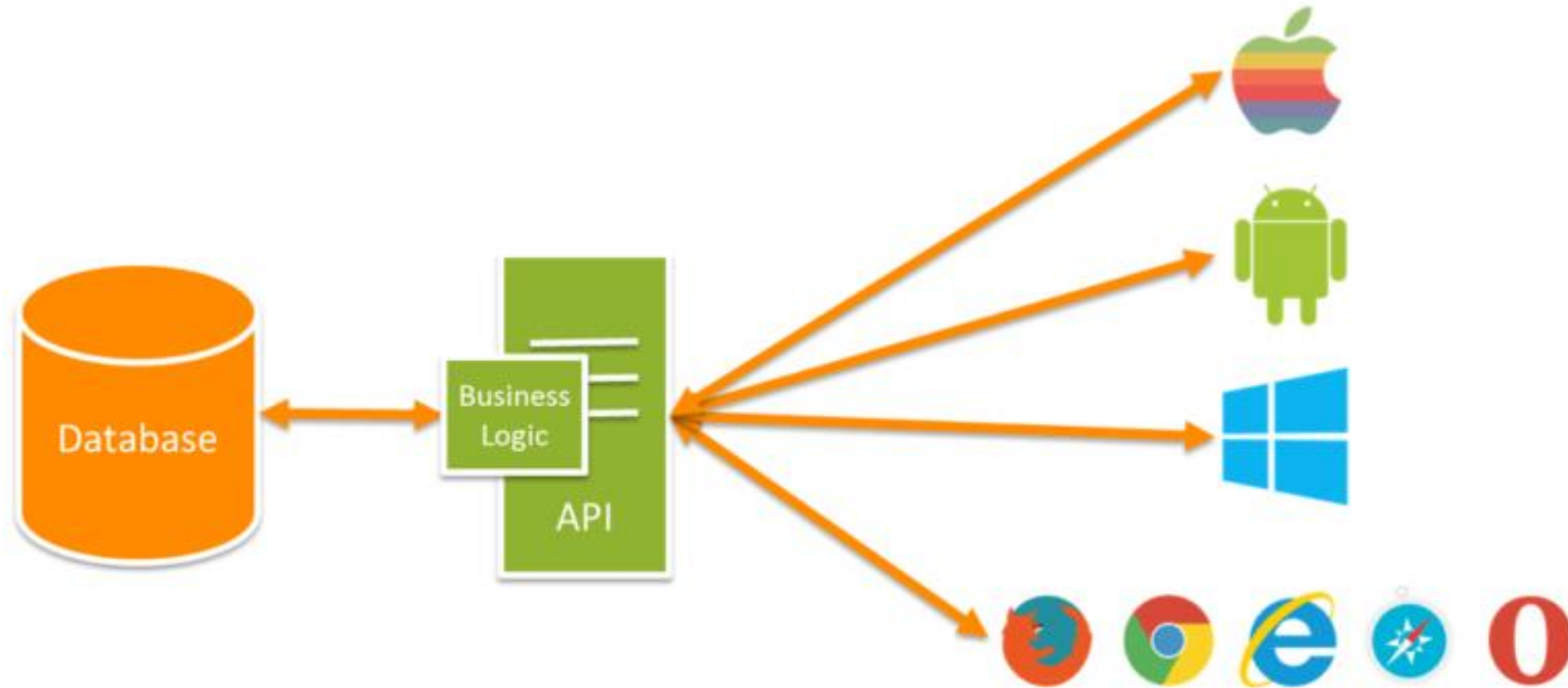
4 Python Best Practices

1. **Reproducibility** -> Your code should be reproducible and the best way to ensure that's the case is providing it alongside the virtual environment requirements to run it. We achieve this via poetry.
2. **Black** -> opinionated formatter. *"Blackened code looks the same regardless of the project you're reading. Formatting becomes transparent after a while and you can focus on the content instead."*
3. **Flake8** -> Linter. *"The wrapper which verifies pep8, pyflakes, and circular complexity"*
4. **Mypy** -> Type-checker. *"Python is a dynamic language, so usually you'll only see errors in your code when you attempt to run it. Mypy is a static checker, so it finds bugs in your programs without even running them!"*
5. **Unittesting** -> Code should be structure in such a way that each functionality is held within a function that is easily testable. **A hard-to-test code is a big symptom of bad code!**
6. **Automatic checks and testing** -> In a real project, you want to be able to automatically build a environment that replicates the local one where all the checks and tests are run: linting, type-checking and unittests (and integration tests if there were).

Connecting to APIs

1 What's an API?

1. REST API is the most used one but not the only API architecture.
2. REST defines a set of functions like GET, PUT, DELETE, etc. that clients can use to access server data. Clients and servers exchange data using HTTP. **The main feature of REST API is statelessness.**



Connecting to APIs

2 Connecting to APIs in python

1. We will use *requests* library.
2. Things to consider when working with APIs: (1) authentication, (2) status code, (3) timeouts and connection errors, (4) rate limits, (5) response schema (<https://json-schema.org/learn/miscellaneous-examples.html>)

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
'{"type":"User"... '
>>> r.json()
{'private_gists': 419, 'total_private_repos': 77, ...}
```

Homework

Q&A