

Programação II

Apresentação da cadeira

Hugo Pacheco

DCC/FCUP

21/22

Programação

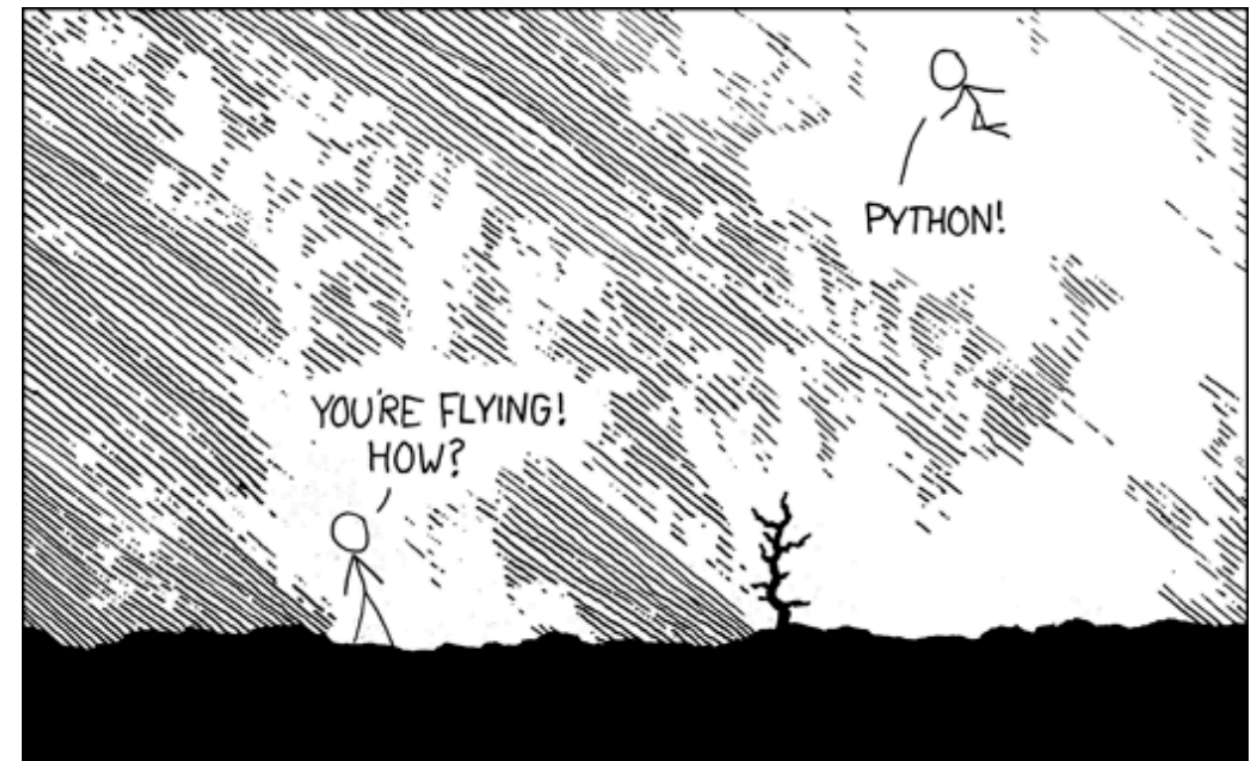
- Programação =
 - Automação de processos
 - Resolução de problemas
 - Extração de informação
 - Visualização de informação

Programação I

- Introdução à programação e aos computadores
- Conceitos base de programação em Python
- Apresentação mais **bem-fundada**:
 - Partir dos fundamentos de programação: funções, listas, dicionários, classes, exceções, etc
 - Escrever programas simples que ilustram os conceitos
 - Aprender programação como resolução de problemas

Python

- Uma linguagem de alto nível e de alta produtividade
- “batteries included”: milhares de bibliotecas prontamente disponíveis
 - gráficos
 - processamento de dados
 - web
 - ...



Programação II

- Cadeira profundamente prática de programação em Python
- Apresentação mais **pragmática**:
 - Aprender a utilizar alguns dos melhores recursos de programação para manipulação automática de informação
 - Desenvolver familiaridade com as bibliotecas e ambiente de desenvolvimento existentes e melhorar experiência de programação
 - O objetivo não é formar especialistas. Tanto quanto possível, não vamos nem queremos olhar para dentro do “capô”

Análise de dados

- É uma fração significativa de toda a programação que é feita hoje em dia
- É a principal forma de programação feita por programadores não profissionais
- Python é uma linguagem bastante forte neste domínio
- Pode ser aplicada em domínios variados, não necessariamente por especialistas de informática: biologia, ciências sociais, medicina, geografia, economia, etc
- “Data-driven world”: grandes quantidades de dados, necessidade de automatização e extração de conhecimento

90% of the world's data
was generated over the last
2 years...



Programa

1. Revisão dos conceitos base

- números, strings, listas, dicionários, classes, etc

2. Análise de dados

- formatos comuns de partilha de informação
- modelos comuns de representação e manipulação de dados

3. Visualização de dados

- diferentes formas de visualização de dados (gráficos, mapas)

4. Animações e aplicações interativas

- Construção de aplicações interativas para visualização de dados

Avaliação

- Não serão registadas presenças (teóricas e práticas)
- Avaliação completamente prática
- **70%** nota prática:
 - 3 trabalhos práticos **individuais** para serem desenvolvidos nas aulas práticas e em casa
 - entrega até ~15 dias depois da última aula prática sobre o tema (excepto último trabalho)
- **30%** nota oral: defesa individual do trabalho no fim do semestre
 - elegância, organização e documentação do código
 - capacidade de resposta a perguntas sobre o trabalho
 - componente contínua (acompanhei o vosso trabalho nas aulas práticas)
- É possível melhorar e entregar os trabalhos em época de recurso, mas será exigida nova funcionalidade; têm que ter frequência em época normal

Trabalhos práticos

- O enunciado de cada trabalho estará dividido em Tarefas.
- Podem utilizar os enunciados do ano passado como exercícios adicionais para treinar
- Cotações de cada Tarefa serão decididas mais à frente consoante o progresso das aulas
- Algumas Tarefas, quando indicado, poderão servir como valorização (a decidir consoante o progresso das aulas)
- Esta não tenciona ser uma cadeira técnica: os principais critérios de avaliação serão **esforço** e **interesse**

Trabalhos práticos

1. Análise de texto

- processamento do texto do *Sermão de Santo António aos Peixes* do Padre António Vieira: contar palavras, agrupar, reformatar, etc

2. Análise de dados

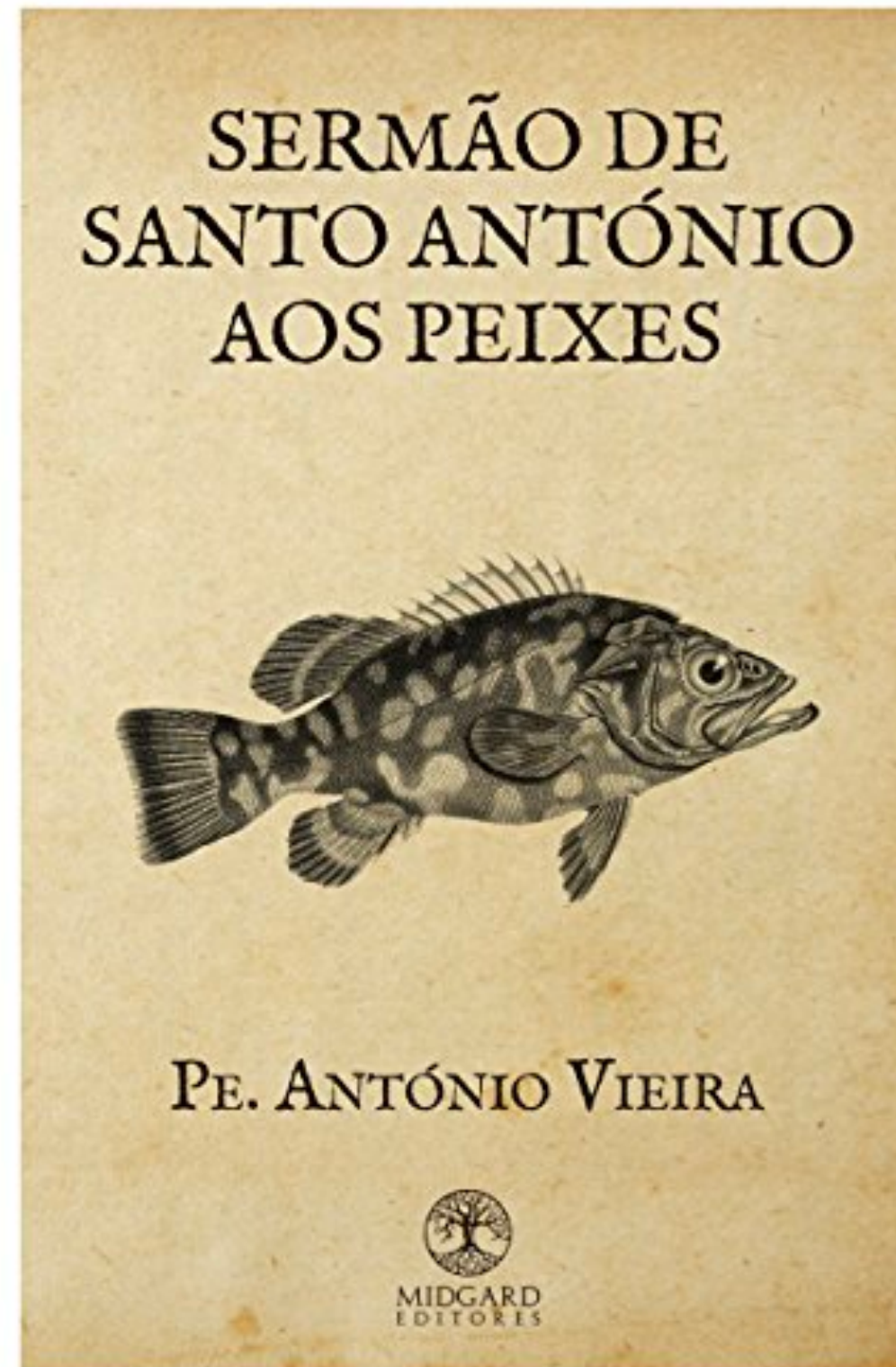
- análise de dados fornecidos pelo IPMA: previsão meteorológica, sismicidade, etc

3. Visualização e animação de dados

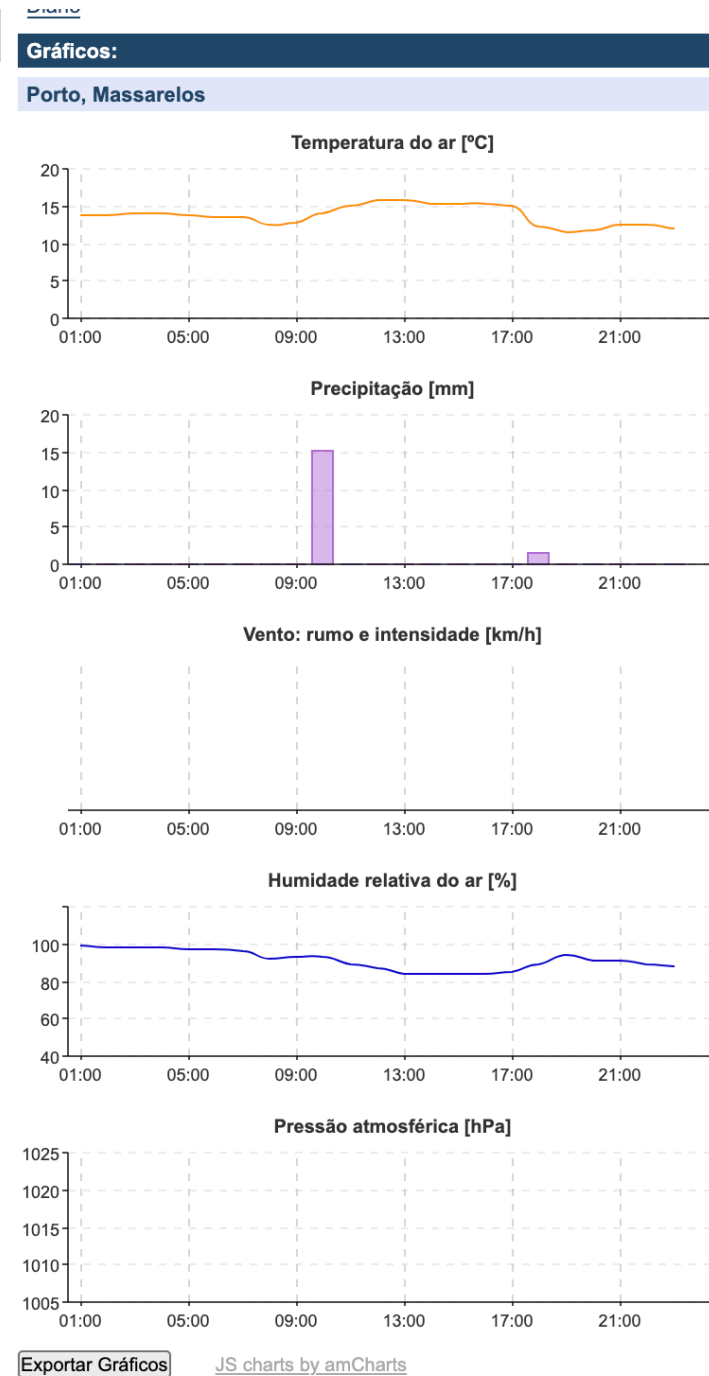
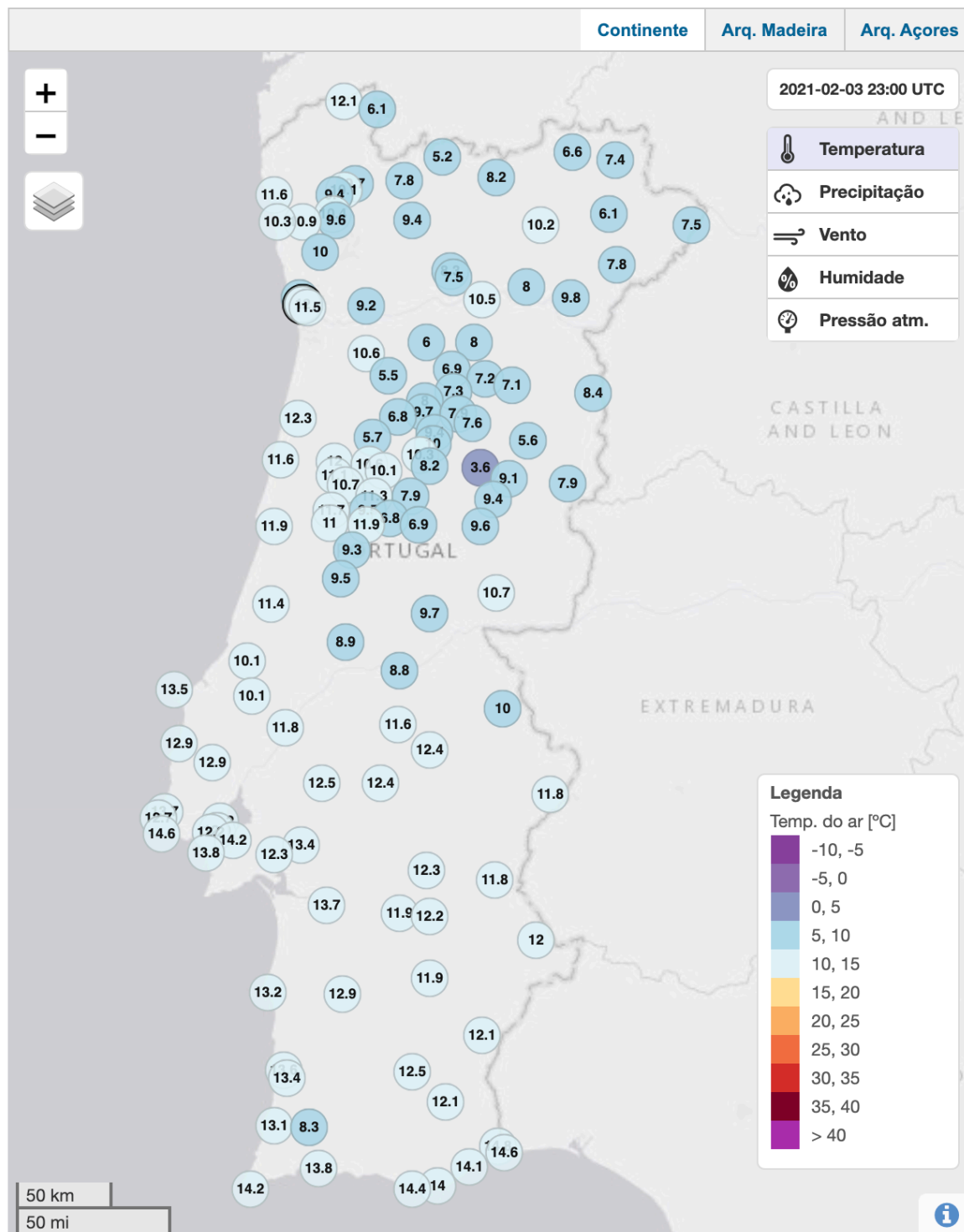
- análise e visualização de dados sobre a pandemia COVID-19 publicados pela DGS

Nota: Temas ainda sujeitos a alteração

Projeto 1

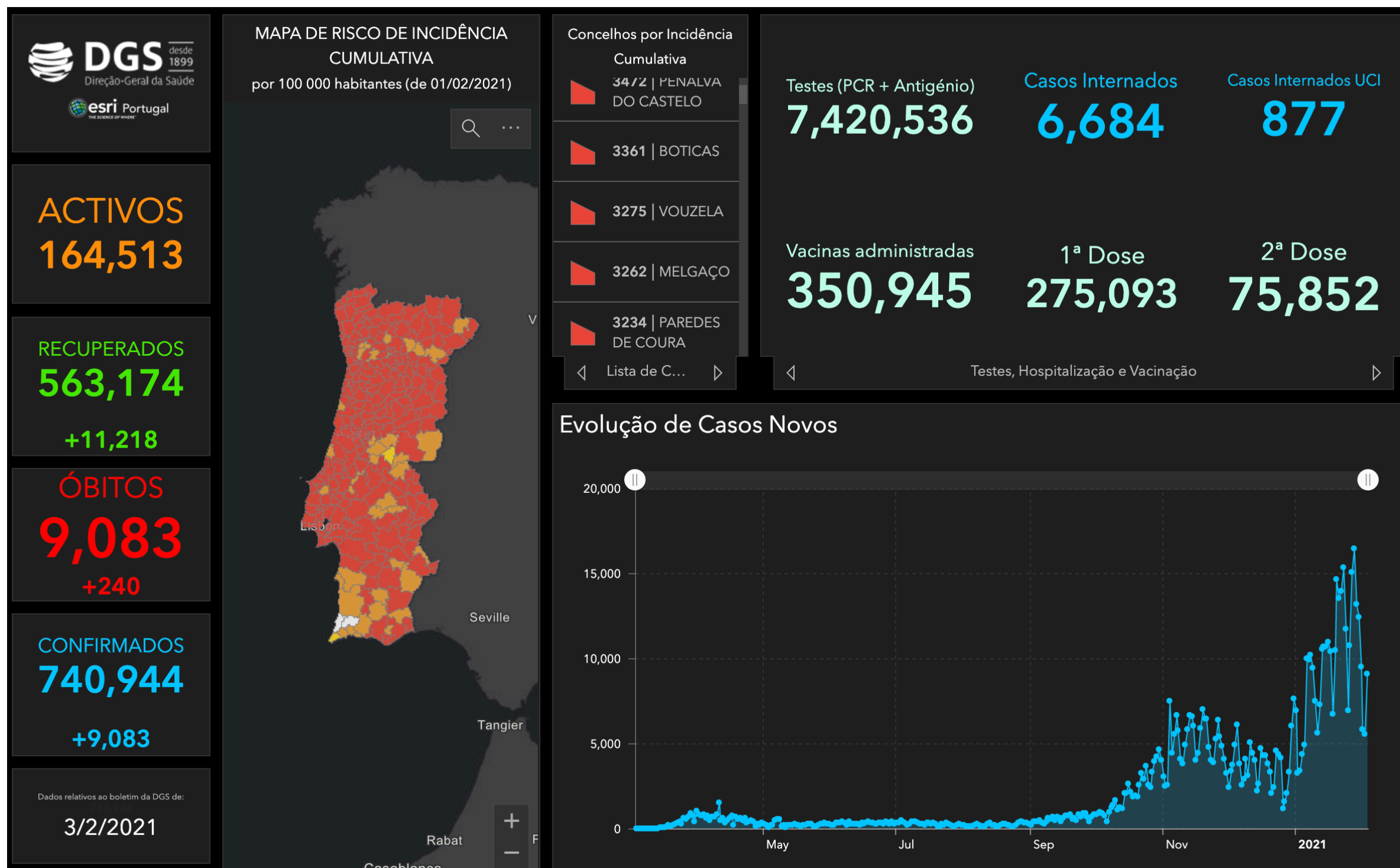


Projeto 2



Representação gráfica dos parâmetros observados para as últimas 24 horas na estação selecionada no mapa.

Projeto 3



Entregas

- Entrega por email para o docente (hpacheco@fc.up.pt)
 - Enviar apenas um ficheiro de código com o nome projeto?.py
 - Não enviar ficheiros de suporte (texto, json, imagens, gráficos, ...)
 - Não alterar o template (assinaturas de funções e código pré-definido)
- Datas **provisórias**, podem vir a ser alteradas com o calendário
 - Projeto 1: entrega até 3 Abril
 - Projeto 2: entrega até 15 Maio
 - Projeto 3: entrega até 07 Junho

Informações Gerais

- Sigarra: site oficial da cadeira
- Github: repositório com todo o material de suporte à cadeira (slides, projetos práticos, etc)
- Teams: plataforma para comunicação fora das aulas; instant messaging

Aulas

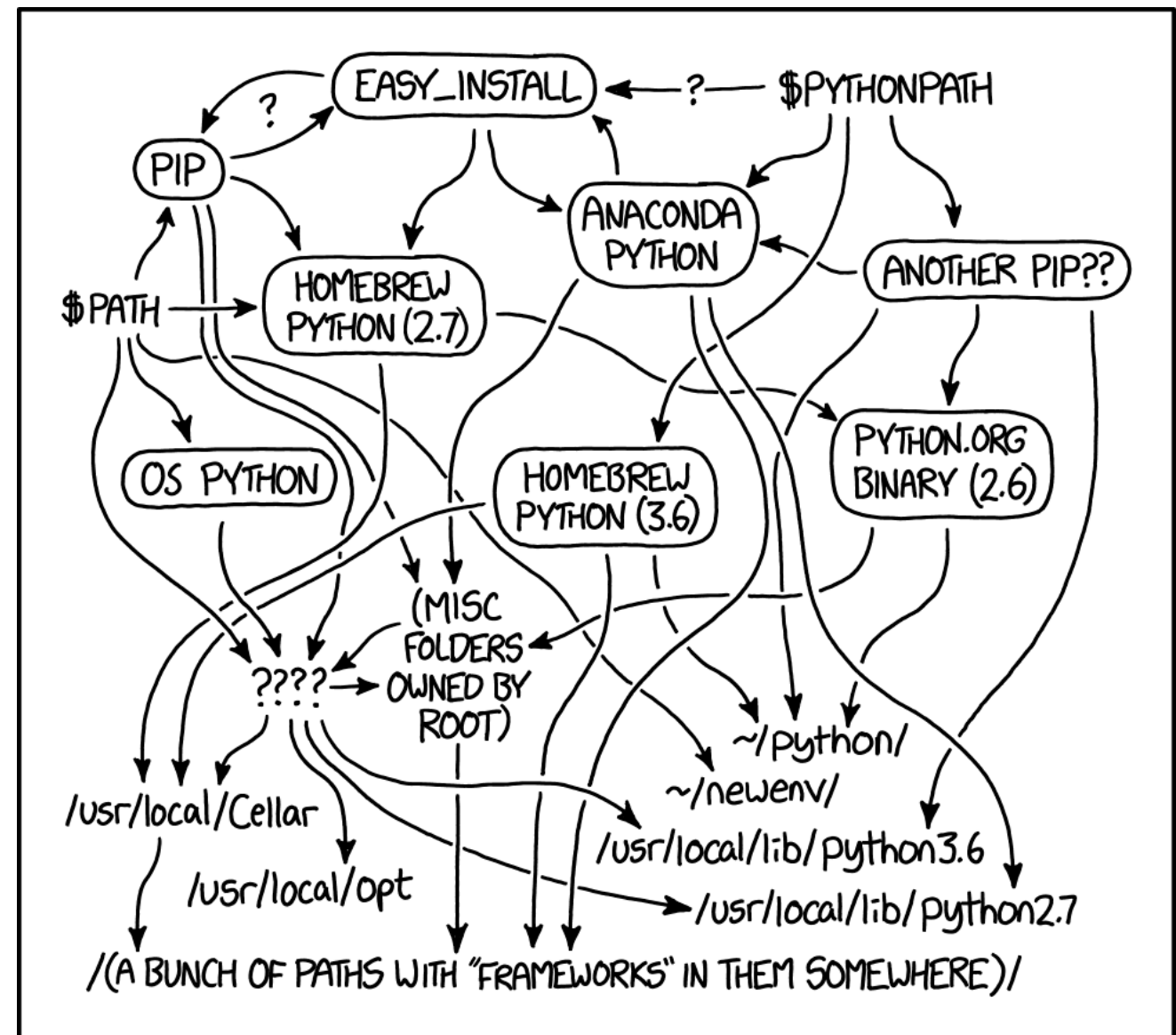
- Em regime **presencial**
- Aulas teóricas
 - apresentação da matéria e exemplos práticos de utilização
 - vídeos gravados do ano anterior, conteúdo essencialmente igual
- Aulas práticas
 - resolução dos guiões dos projetos práticos

Docentes

- Hugo Pacheco (hpacheco@fc.up.pt):
- Não existe horário de atendimento pelo menos por enquanto. Apela-se à comunicação pelo **Teams** ou por email institucional

TPC: Instalar o Python

- Existem muitas formas de instalar o Python e bibliotecas associadas: *binário, anaconda, pip, brew, apt-get*, etc
- Se já estão familiarizados com um ambiente, usem esse; evitem ter múltiplas instalações do Python



TPC: Instalar o Python

- Como sugestão...
- Instalar Python 3.x “nativo”
 - Windows: fazer download do instalador oficial em <https://www.python.org/downloads/>
 - Linux: *sudo apt-get install python3*
 - Mac OS: *brew install python3*
- Utilizar o editor PyCharm CE (<https://www.jetbrains.com/pycharm/download/>)
 - code highlighting, debugging, documentação, etc
 - gestão e instalação automática de pacotes, não interfere com outras possíveis instalações

Projeto 0 - Desenferrujar

- Primeiras duas semanas de aulas práticas para rever matéria e aferir conhecimentos
- Enunciado do Projeto 0
- Antes da aula:
 - Instalem o Python
 - (Alternativa) editor online replit