**INSTITUTO SUPERIOR DE CONTABILIDADE E ADMINISTRAÇÃO DE COIMBRA**

**Curso:** LICENCIATURA EM CIÊNCIA DE DADOS PARA GESTÃO

**Unidade Curricular:** PROJETO EM CIÊNCIA DE DADOS

**Ano Letivo:** 2025/2026

**REGRAS DE OURO:**

1. **"SE NÃO ESTÁ NO *GITHUB*, NÃO ACONTECEU"**
2. **MENSAGENS DE *COMMIT* DESCRITIVAS**
3. **README VISUALMENTE APELATIVO**
4. **REPOSITÓRIO *GUITHUB* ORGANIZADO E LIMPO**

# *MILESTONE 1 – INICIAÇÃO (SEMANAS 1-3)*

**Foco (CRISP-DM):** ***Business Understanding & Data Understanding***

**Semana 1:** Formação dos grupos (2-3 alunos) e escolha do tema da lista fornecida pelo docente. Revisitar a metodologia CRISP-DM.

**Semana 2:** (*Business Understanding*) Configuração do *GitHub* (convite ao professor como colaborador). Definição do problema, objetivos e critérios de sucesso. Criação do *README.md* inicial.

**Semana 3:** (*Data Understanding*) Primeira exploração dos dados no *Kaggle Code*. Carregamento dos dados, análise de tipos de dados, valores nulos e estatística descritiva básica. Documentar *docs/M1_iniciacao.md*.

## A. *Checklist* Detalhado: *Milestone* 1 (Semanas 1-3)

**Semana 1: Formação de Equipas de Trabalho**

Foco: Organização de equipas

Aula 2 (06-02-2026): Constituição

[ ] Grupos de 2-3 alunos formados e comunicados ao professor.

**Semana 2: Setup Técnico e *Business Understanding* (O Problema)**

Foco: Preparação da infraestrutura e definição do propósito do projeto no ficheiro *docs/M1_iniciacao.md*.

Aula 3 (11-02-2026): Tema e Infraestrutura de Colaboração

[ ] Escolha do conjunto de dados (*dataset*) a partir da lista disponibilizada.

[ ] Leitura preliminar da documentação do *dataset* (no *Kaggle*) para entender as variáveis.



Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt

[ ] Criação do repositório no *GitHub* com o nome do projeto.
[ ] Adição de todos os elementos do grupo e do professor como colaboradores.
[ ] Criação da estrutura de pastas definida (focando em .gitignore, README.md, e pasta docs/).
[ ] Realização do primeiro *commit* por cada membro (para testar permissões).

### Aula 4 (13-02-2026): Definição de Objetivos, Planeamento e Regras

[ ] Escrita da Introdução e Contexto (O que é o tema? Porque é relevante?).
[ ] Definição dos Objetivos SMART (O que queremos prever? Qual a métrica de sucesso?).
[ ] Criação das Perguntas de Investigação (Pelo menos 3 perguntas que os dados devem responder).
[ ] Preenchimento da Divisão de Papéis inicial entre os membros do grupo.
[ ] Identificação das ferramentas e bibliotecas *Python* que pretendem usar.
[ ] Primeira versão do README.md (resumo do projeto e links para o *Kaggle*).

**Semana 3: *Data Understanding & Validação***

Foco: Inspeção inicial dos dados e fecho da *Milestone* 1.

### Aula 5 (18-02-2026): Inspeção Inicial (*Kaggle*)

[ ] Importação do *dataset* para um Notebook no *Kaggle*
[ ] Execução de comandos básicos (*df.info()*, *df.describe()*, *df.head()*).
[ ] Verificação de integridade: Existem valores nulos? Os tipos de dados (*int*, *float*, *object*) estão corretos?
[ ] Documentação destas primeiras impressões no ficheiro *docs/M1_iniciacao.md* (Secção de Descrição Técnica).

### Aula 6 (20-02-2026): Fecho da *Milestone* 1 e Submissão até 24-02-2026

[ ] Revisão final de todos os campos do *template docs/M1_iniciacao.md*.
[ ] Garantia de que todos os ficheiros do *kaggle* foram enviados para o *GitHub* (*git push*).
[ ] Verificação se as mensagens de *commit* seguem as boas práticas (p/ex.: Docs: Finalização da *Milestone* 1).
[ ] *Check-out* com o Professor: Breve apresentação ao professor para validar a viabilidade do projeto.

## O que deve estar pronto para Avaliação do *Milestone 1* (24-02-2026)

Até ao final do dia **24 de fevereiro de 2026**, deve gerar o primeiro lançamento (*release*) do portefólio que tem de refletir o seguinte:

- Repositório: Estrutura completa do projeto e membros da equipa associados (incluindo professor)
- Ficheiro *README.md*: A primeira versão preenchida seguindo as recomendações.
- Ficheiro *docs/M1_iniciacao.md*: Totalmente preenchido seguindo o modelo recomendado.
- Ficheiro *.gitignore*: Configurado (para não subir pastas como *__pycache__* ou ficheiros *.DS_Store*).


Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt

- Histórico de *Commits*: Deve haver atividade equitativa de todos os membros do grupo.
- *Kaggle*: Um *notebook* inicial já ligado ao *dataset*, com os dados carregados e visualizados, e respetiva versão guardada no *GitHub*.

## B. Guia de Iniciação: *GitHub* e Estrutura de Projeto

Configuração do ambiente de trabalho no *GitHub* para a Unidade Curricular, garantindo que o projeto segue padrões profissionais de mercado.

### Criar e Configurar a Conta *GitHub*

1. Registo: Aceda a github.com e crie uma conta. Dica: Use o seu *email* institucional.
2. Perfil: Adicione o seu nome real e uma fotografia. Na Ciência de Dados, o seu *GitHub* funciona como um currículo vivo.
3. Criar Repositório:

   Clique no botão "+" (canto superior direito) -> *New repository*
   *Repository name*: Escolha um nome curto e claro (p/ex.: projeto-cd-grupo1)
   *Descrition*: Faça uma pequena descrição do projeto
   *Public/Private*: Defina como *Public*
   *Initialize this repository with*: Selecione *Add* a *README file*

### Criar a Estrutura de Pastas (Sem usar a linha de comandos)

Se ainda não domina o terminal, pode criar a estrutura diretamente no *browser*, de acordo com a estrutura obrigatória indicada no ponto E:

1. No seu repositório, clique em *Add file* -> *Create new file*.
2. Para criar pastas e subpastas, escreva o nome da pasta seguido de uma barra /.
   Exemplo: Escreva *data/raw/.gitkeep* para criar a pasta *data* e a subpasta *raw*.
   Nota: O ficheiro *.gitkeep* é apenas um ficheiro vazio para "forçar" o *GitHub* a manter a pasta criada.

## C. Como trabalhar em equipa no *GitHub*

1. Convites: No repositório, vá a *Settings* -> *Collaborators* -> *Add people* e adicione os seus colegas de grupo, usando o *email* das respetivas contas *GitHub*, bem como o docente usando o *email* dmelo@iscac.pt
2. *Commits*: Sempre que algum elemento do grupo fizer uma alteração importante (p/ex.: terminar a limpeza de dados), faz um "commit" com uma mensagem clara:

   ✗ Mensagem má: "*Update*"
   ✓ Mensagem boa: "*feat*: limpeza de valores nulos na coluna idade"



Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt

3. *Docs*: Use a pasta *docs/* para escrever e clarificar todo o processo e raciocínio de desenvolvimento do projeto. O código diz como foi feito, a documentação diz porque foi feito.
4. Dica Importante: O repositório é o vosso portefólio. Mantenham-no organizado, limpo e com um README visualmente apelativo. É no repositório que o docente verá o vosso progresso real em Ciência de Dados.

## D. Estratégia de *Freeze* (*GitHub Releases*)

Para garantir a versão exata do portefólio submetido para avaliação na data limite de cada *Milestone* (mesmo que se continue a trabalhar no projeto no dia seguinte), deve usar a opção *Releases*:

1. No *GitHub*, selecione *Releases -> Create a new release*.
2. No *Tag version*, identifique a nova versão (p/ex.: v1.0-M1 para a *Milestone 1*).
3. Atribua um título adequado (p/ex.: "Entrega Final *Milestone 1*").
4. Isto cria uma imagem exata (*snapshot*) do seu projeto naquela data e hora exata, salvaguardando essa versão e permitindo que continue a trabalhar no projeto.

## E. Estrutura Obrigatória do Repositório *GitHub* (Adaptada ao Planeamento)

O repositório deve ter exatamente a seguinte organização para garantir a nota de organização (*Milestone* 1).

```
nome-do-projeto/
├─ .gitignore            # Ficheiros a ignorar (p/ex.: caches do Python, ficheiros temporários)
├─ README.md             # Relatório Principal: O resumo executivo de todas as fases
├─ requirements.txt      # Bibliotecas (pandas, scikit-learn, etc.) usadas no Kaggle
│
├─ data/                 # Gestão de Dados (amostras ou versões processadas)
│   ├─ raw/       # Cópia (se pequena) ou link para o dataset original do Kaggle
│   └─ processed/ # Dados após a fase de Data Preparation (limpos)
│
├─ docs/                 # O DIÁRIO DE BORDO - Markdowns (.md)
│   ├─ M1_iniciacao.md   # Business Understanding: Objetivos, escolha do dataset e planeamento
│   ├─ M2_exploracao.md  # Data Understanding & Preparation: Insights da EDA e decisões de limpeza
│   ├─ M3_modelacao.md   # Modeling & Evaluation: Comparação de modelos e métricas finais
│   └─ M4_conclusoes.md  # Deployment & Reflexão: O que o projeto resolveu e limitações
│
├─ notebooks/                          # Versões exportadas do Kaggle Code
│   ├─ 1.0_eda_limpeza.ipynb           # Corresponde à Fase 2
│   ├── 2.0_modelacao_treino.ipynb     # Corresponde à Fase 3
│   └── 3.0_interpretacao.ipynb        # Corresponde à Fase 4
│
├─ src/                          # Código modular (Opcional para projetos mais simples)
│   ├─ data_loader.py            # Funções para leitura e tratamento inicial
│   └─ evaluation_utils.py       # Funções para gerar matrizes de confusão/gráficos de erro
│
└─ reports/              # Evidências Visuais
    ├─ figures/          # Gráficos (PNG/JPG) exportados para ilustrar o README
    └─ apresentacao.pdf  # Documento final ou link para os slides da Semana 15
```



Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt

## F. Ficheiro *.gitignore* – Recomendado (Copiar e Colar)

Para que o repositório não guarde lixo, nomeadamente ficheiros temporários ou dados demasiado pesados, é necessário criar um ficheiro chamado *.gitignore* na raiz com o seguinte conteúdo:

```
# --- Python & Ambientes ---
__pycache__/
*.py[cod]
*$py.class
.venv/
venv/
env/

# --- Kaggle Especifico ---
# Ignorar chaves de API privadas (nunca partilhar o kaggle.json!)
.kaggle/
kaggle.json

# --- Jupyter & Notebooks ---
.ipynb_checkpoints/
*/.ipynb_checkpoints/*

# --- Gestão de Dados (Estrutura UC) ---
# Ignorar os dados reais para não ultrapassar o limite de 100MB do GitHub.
# Os dados devem estar no Kaggle ou num link externo.
data/raw/*
data/processed/*

# Mas manter as pastas no GitHub (usando o truque do .gitkeep)
!data/raw/.gitkeep
!data/processed/.gitkeep

# --- Sistema & IDEs ---
.DS_Store
.vscode/
.idea/
*.log
```

## G. Modelo de README.md

O ficheiro *README.md* é o documento que permite apresentar um resumo executivo do projeto e deve ser guardado na raiz do projeto.

```
# [Título do Projeto de Ciência de Dados]

## Identificação da Equipa
* **Grupo nº:** [Número]
* **Membros:**
  * [Nome do Aluno 1] - [Nº de Estudante]
  * [Nome do Aluno 2] - [Nº de Estudante]
  * [Nome do Aluno 3] - [Nº de Estudante]



## Organização do Repositório

A estrutura deste projeto segue as boas práticas de Ciência de Dados e Engenharia de Software:

* **`data/`**: Armazenamento de dados (dados brutos em `raw/` e processados em `processed/`).
* **`docs/`**: Documentação técnica detalhada dividida por Milestones (M1, M2 e M3).
* **`notebooks/`**: Jupyter Notebooks para experimentação, limpeza e modelação.
```



Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt

```
* **`src/`**: Código-fonte modular (scripts `.py`) para funções reutilizáveis.
* **`reports/`**: Relatórios finais, apresentações e exportação de figuras (`figures/`).
* **`requirements.txt`**: Ficheiro de configuração com as bibliotecas necessárias.



## 1. Iniciação (Milestone 1)
### Contexto e Problema de Negócio
[Descreve o problema que pretendem resolver. Qual é o desafio da empresa ou organização?]

### Objetivos do Projeto
* **Objetivo 1:** [Ex: Prever a rotatividade de clientes]
* **Objetivo 2:** [Ex: Identificar os principais fatores de influência]

### Fonte de Dados
* **Dataset:** [Link para a fonte ou descrição dos ficheiros]
* **Dimensão:** [Ex: 10.000 linhas, 15 colunas]



## 2. Exploração (Milestone 2)
### Limpeza e Preparação
* [Breve resumo das ações de limpeza tomadas. Detalhes em `docs/M2_exploracao.md`]

### Principais Conclusões (EDA)
> *Dica: Insere aqui o gráfico mais importante do projeto.*
* **Ponto-chave:** [Ex: Identificámos que o fator X influencia em 40% o resultado Y, por aplicação
do método ganho de informação]



## 3. Modelação (Milestone 3)
### Abordagem Técnica
* **Modelos:** [Ex: Random Forest e XGBoost]
* **Métrica Principal:** [Ex: F1-Score ou RMSE]



## 4. Finalização (Milestone 4)
### Resposta ao Problema
[Resumo da solução e como ela gera valor para o negócio.]

### Recomendações de Inovação
1. [Sugestão prática baseada nos resultados]



## Como Reproduzir este Projeto
1. Clone o repositório: `git clone [url-do-repo]`
2. Instale as dependências: `pip install -r requirements.txt`
3. Execute os notebooks na pasta `notebooks/` seguindo a ordem numérica.


**Instituição:** Coimbra Business School | ISCAC
**Curso:** Licenciatura em Ciência de Dados para a Gestão
**Unidade Curricular:** Projeto em Ciência de Dados
**Professor Responsável:** Dora Melo (dmelo@iscac.pt)
```



Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt

## H. Modelo de M1_iniciacao.md

O ficheiro *M1_iniciacao.md* é o documento que deve detalhar o planeamento estratégico que servirá de base para todo o desenvolvimento do projeto, ou seja, consiste no plano de trabalho detalhado. Este ficheiro deve estar guardado na pasta *docs/* do projeto.

```
# Milestone 1: Iniciação e Definição do Projeto

## 1. Descrição Detalhada do Problema
[Expandir a descrição do README. Explicar o contexto do setor (ex: retalho, banca, saúde) e porque é que este problema é relevante no momento atual.]

## 2. Objetivos SMART
*Defina os objetivos do projeto seguindo a lógica SMART (Específico, Mensurável, Atingível, Relevante e Temporal):*

1.  **Objetivo 1:** [Ex: Reduzir o erro de previsão de stock em 15% até ao final do semestre.]
2.  **Objetivo 2:** [Ex: Identificar os 5 principais perfis de consumo através de técnicas de clustering.]

## 3. Metodologia de Gestão (PBL)
* **Divisão de Tarefas:**
  * **Membro A:** Responsável pela Engenharia de Dados.
  * **Membro B:** Responsável pela Modelação Estatística.
  * **Membro C:** Responsável pela Visualização e Documentação.
* **Ferramentas de Colaboração:** [Ex: GitHub Projects para Kanban, reuniões semanais via Teams/Discord].

## 4. Análise de Viabilidade dos Dados
* **Disponibilidade:** [Os dados já foram descarregados? Estão em base de dados?]
* **Qualidade Inicial:** [Ex: Notámos que faltam dados de datas em algumas colunas, precisaremos de tratar isso na M2.]
* **Ética:** [Os dados cumprem o RGPD? Estão anonimizados?]

## 5. Cronograma Interno
| Fase | Data Limite | Entregável Esperado |
| :--- | :--- | :--- |
| M1: Iniciação | [Data] | Repositório estruturado e Plano de Projeto. |
| M2: Exploração | [Data] | Notebook de EDA e Dados Processados. |
| M3: Modelação | [Data] | Comparação de algoritmos e métricas. |
| M4: Finalização| [Data] | Pitch e Relatório Final. |

---
*Data de última atualização: [DD/MM/AAAA]*
```

## I. Como Integrar o *Kaggle* com o *GitHub*

O *Kaggle* não permite definir uma pasta local sincronizada automaticamente com o repositório *GitHub*. O fluxo de atualização do portefólio *GitHub* recomendado é:

1. Desenvolvimento: escrita do código no *Kaggle*.
2. Sincronização: No menu *File* do *Kaggle Notebook*, selecione a opção *File -> Link to GitHub*. Esta ação permite fazer um *Save Version* e que o *Kaggle* envie automaticamente uma cópia do *notebook* para o seu repositório *GitHub*.



Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt

Ao selecionar *Link to GitHub* dentro do *Kaggle*, é aberta uma janela de autenticação do *GitHub*. O *login* deve ser feito com as credenciais da conta *GitHub* do seu portefólio, independentemente do *email* da conta do Kaggle.

3. Organização: Garanta que, ao vincular, define o caminho correto para a pasta */notebooks* do seu projeto/portefólio, bem como a atribuição de um nome adequado à versão do código.

### O que é enviado (e o que vê no GitHub)

- Código e *Markdown*: Sempre enviados.
- Gráficos Estáticos: Imagens geradas por bibliotecas como *Matplotlib* ou *Seaborn* são codificadas dentro do ficheiro e aparecem perfeitamente na pré-visualização do *GitHub*.
- Tabelas (*DataFrames*): Os resumos que são gerados (p/ex.: *df.head()*) também são guardados.
- *Logs* de Erro: Se o código falhou e foi guardado assim, o erro também aparecerá no *GitHub*.

### "Save & Run All" vs. "Quick Save"

- *Save & Run All* (*Commit*): O *Kaggle* executa o código do início ao fim, gera os gráficos e guarda-os no ficheiro - é esta versão que deve ir para o *GitHub*.
- *Quick Save*: Se usar esta opção sem ter executado o código integralmente, o ficheiro pode ir apenas com o código, e quem abrir o *GitHub* não verá quaisquer gráficos ou resultados.

### O Problema dos Gráficos Interativos

Se utilizar bibliotecas como *Plotly*, *Bokeh* ou *Altair* (que geram gráficos onde é possível passar o rato e ver valores), o *GitHub* não os consegue replicar.

- No *GitHub*: aparecerá apenas um espaço em branco ou uma mensagem de erro.
- Solução: Para avaliação no *GitHub*, é sempre mais seguro usar gráficos estáticos ou guardar uma versão em imagem (.png) na pasta *reports/figures/*, como planeado na estrutura do projeto.

### Atenção ao Tamanho do Ficheiro

Além do código, como os gráficos e os dados também são guardados "dentro" do *Kaggle notebook*, o ficheiro pode ficar muito pesado.

- Dica: Se gerar 50 gráficos de alta resolução num único *notebook*, o ficheiro *.ipynb* pode chegar aos 20MB ou 30MB. O *GitHub* lida bem com isso, mas torna a navegação lenta para quem visualiza o portefólio.
- Boa Prática: No final do projeto, faça um *Clear All Outputs* num *notebook* de rascunho e mantenha apenas os *outputs* essenciais na versão *notebook* que vai para a pasta */notebooks* do *GitHub*.

## J. Conceitos Importantes

- **O que é um *Commit*?**



Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt

Um *Commit* é entendido como um "Ponto de Restauro" ou um "Save Game" no projeto.
Quando se faz um *commit*, o *Git* grava o estado atual dos ficheiros (como uma imagem), definindo um ponto de restauro. Cada *commit* vem acompanhado de uma Mensagem de *Commit*, onde se explica brevemente o que foi alterado (p/ex.: "Adicionado gráfico de barras na *Milestone* 2").
Por que é importante?

1. Histórico: Se o código deixar de funcionar, é possível recuperar a versão anterior (voltar ao commit), onde tudo estava bem.
2. Responsabilidade: No trabalho de grupo, o *GitHub* mostra quem fez qual *commit*. Assim, o docente sabe exatamente quem contribuiu com o quê.

- **O que é um *Markdown*?**

O *Markdown* é uma forma muito simples de formatar texto usando apenas símbolos do teclado. É a linguagem oficial para escrever documentação no *GitHub*. São os ficheiros com extensão ".md".
Em vez de se usar as opções de "Negrito" ou "Itálico" como no Microsoft Word, escrevem-se códigos simples que o *GitHub* depois transforma em texto bonito.
Exemplos Rápidos:

- Para fazer um Título, usa-se o cardinal: # Título
- Para colocar em Negrito, usam-se asteriscos: **texto em negrito**
- Para criar uma Lista, usa-se um traço: - Item 1
- Para inserir um *Link*: [Nome do Site](url-do-site)

- **O que é o *Git*?**

O *Git* é um Sistema de Controlo de Versões Distribuído (DVCS).
As 3 Funções Principais do *Git*:

1. Rastreabilidade Histórica - O *Git* guarda o histórico completo de todas as alterações feitas num projeto. Se acidentalmente for apagado um código que funcionava na *Milestone* 2, o *Git* permite "recuar no tempo" e recuperar essa versão exata sem perder o que foi feito depois.
2. Trabalho Colaborativo (*Branching*) - Num projeto de grupo, o *Git* permite que vários membros da equipa trabalhem no mesmo ficheiro ao mesmo tempo sem sobreporem o trabalho uns dos outros. Isto é feito através de ramos (*Branches*): um elemento da equipa trabalha na limpeza de dados num ramo; o outro elemento da equipa trabalha na visualização noutro ramo; no final, o *Git* ajuda a fundir (*merge*) estas duas partes no projeto principal.
3. Integridade e Segurança - Como é um sistema distribuído, cada membro da equipa tem uma cópia completa do histórico do projeto no seu computador. Se o servidor falhar, o projeto não se perde. Além disso, cada alteração é verificada por um algoritmo (SHA-1), o que impede a corrupção oculta de ficheiros.

No contexto do projeto, o *Git Local* tem três estados ou áreas principais:


Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt

1. *Working Directory*: pasta onde se editam os ficheiros (p/ex.: a escrever no *Markdown*) e estão guardados fisicamente. Para o *Git*, estas alterações são consideradas modificadas (*Modified*), mas ainda não estão gravadas no histórico.
2. *Staging Area (Index)*: local onde se preparam os ficheiros a incluir na próxima entrega (*git add*). É uma zona intermédia. Quando se faz *git add*, está a dizer-se ao *Git*: "Prepara estes ficheiros específicos para a próxima fotografia (commit)". É o estado *Staged*.
3. *Local Repository (Local)*: local onde se grava permanentemente a versão no teu PC (*git commit*). Quando se faz *git commit*, as alterações são movidas para a base de dados interna do *Git* (a pasta oculta *.git*). Aqui, o estado é *Committed*. O teu trabalho está seguro e gravado no PC. Se não se fizer *commit*, a alteração não existe oficialmente para o histórico.

O *GitHub* (*Remote Repository*) é o servidor *online*, necessário para partilha com restantes membro e acesso público do conteúdo do projeto. O carregamento *online* é feito utilizando o *git push*. Este passo copia o conteúdo do teu *Local Repository* para o *Remote*.

- **O que é a Metodologia SMART?**

Para um projeto de Ciência de Dados, a metodologia SMART é essencial para evitar que o projeto se torne uma exploração interminável sem resultados concretos.

O acrónimo SMART define os cinco critérios fundamentais que um objetivo deve cumprir para ser considerado bem estruturado:

1. S (*Specific*) – Específico: O objetivo deve ser claro e focado num aspeto concreto dos dados.
   Evitar: "Melhorar o modelo."
   Usar: "Reduzir o erro médio absoluto (MAE) na previsão de preços de imóveis."
2. M (*Measurable*) – Mensurável: Deve ser possível quantificar o progresso através de métricas de desempenho.
   Exemplos: Percentagem de precisão total (*Accuracy*), *F1-Score*, ou tempo de processamento.
3. A (*Achievable*) – Atingível: O objetivo deve ser realista face aos dados disponíveis e ao conhecimento técnico da equipa.
   Pergunta: "O *dataset* tem variáveis suficientes para esta previsão?"
4. R (*Relevant*) – Relevante: Deve estar alinhado com o problema de negócio ou a pergunta de investigação científica proposta.
   Pergunta: "Resolver este problema traz valor ou novos conhecimentos?"
5. T (*Time-bound*) – Temporal: Deve ter um prazo de entrega ou conclusão definido (neste caso, as datas dos *Milestones*).
   Exemplo: "Concluir a análise exploratória até ao final da Semana 4."


Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt

### Exemplos de questões SMART:

- "Desenvolver um modelo de classificação para prever o *churn* de clientes com um F1-Score mínimo de 0.80, utilizando o histórico de compras dos últimos 12 meses, para ser apresentado no *Milestone* 3."
- "Treinar um algoritmo que identifique patologias em imagens de raio-X com uma Sensibilidade (*Recall*) superior a 92%, garantindo que o número de falsos negativos seja minimizado até à entrega final do projeto."
- "Criar um *pipeline* de deteção de anomalias que identifique transações fraudulentas em tempo real, mantendo uma Precisão de pelo menos 85% e um tempo de inferência inferior a 200ms por transação, até à Semana 6."
- "Aplicar técnicas de *Clustering* (*K-Means*) para segmentar a base de dados em 5 perfis de consumo distintos, validando a coesão dos grupos através do Coeficiente de Silhueta (> 0.5), até ao final da Aula 8."
- "Construir um modelo de regressão para prever o valor de mercado de imóveis urbanos, reduzindo o Erro Médio Absoluto (MAE) para menos de 5.000€, utilizando o *dataset* do *Kaggle* até ao *Milestone* 3."

### Exemplos de perguntas de investigação (depende sempre do objetivo e tema de trabalho):

- "Qual é a correlação entre o Índice de Massa Corporal (IMC) e a probabilidade de um diagnóstico positivo de diabetes?"
- "Quais são as 3 variáveis clínicas que mais contribuem para a previsão de um evento cardíaco?"
- "Existe uma relação direta entre o tempo que um utilizador passa na página do produto e a probabilidade de conversão (compra)?"
- "Filmes com maior orçamento têm tendência a receber melhores notas dos críticos no IMDB ou apenas maior faturação?"
- "Quais são as características demográficas mais comuns entre os clientes que entram em incumprimento de crédito (*default*)?"
- "Como é que o histórico de emprego (anos no cargo atual) influencia o montante de crédito aprovado?"

### *Checklist* de Validação: O Vosso Objetivo é SMART?

Verifiquem se o vosso objetivo responde "SIM" a todas as perguntas abaixo:

[ ] Específico (S): Está claro exatamente o que pretendem prever ou analisar? (Evite palavras como "melhorar", "analisar" ou "ver" sem contexto).

[ ] Mensurável (M): Existe uma métrica numérica (p/ex.: %, RMSE, F1-Score, Precisão) para saber se tiveram sucesso?

[ ] Atingível (A): Têm os dados necessários no *dataset* para responder a este objetivo? A equipa consegue implementá-lo no tempo disponível?

[ ] Relevante (R): Este objetivo resolve o problema central do tema que escolheram? É importante para o projeto de Ciência de Dados?

[ ] Temporal (T): Definiram a qual *Milestone* este objetivo pertence?



Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt

Sugestão: Escrevam o vosso objetivo num *post-it* digital no *GitHub Issues/Projects* e usem e a *checklist* para fazer "autocorreção".

- **O que é o *GitHub Projects*?**

O *GitHub Projects* é uma ferramenta de gestão de projetos integrada no *GitHub* que funciona como um quadro visual dinâmico (de notas em *post-its*). De forma simples, é o espaço onde a equipa organiza o "quem faz o quê e quando". Em vez de apenas guardarem código, as equipas utilizam o *Projects* para:
- Visualizar o Fluxo de Trabalho: Através de colunas (p/ex.: "To Do", "In Progress", "Done"), permite ver instantaneamente o estado de progresso do projeto.
- Centralizar Tarefas (*Issues*): Cada tarefa ou objetivo é um "cartão" que pode conter *checklists*, responsáveis atribuídos, prazos e comentários técnicos.
- Sincronização Direta: Como está dentro do repositório, pode-se associar o código que está associado diretamente à tarefa que o motivou.

Como criar "post-its" e a usar as *checklists* de autocorreção:

1. Criar o Quadro de Projeto
   - No repositório do grupo, cliquem no separador "Projects" (no topo).
   - Cliquem em "New Project" e selecionem o modelo "Board".
   - Deem o nome de, por exemplo, "Planeamento Milestone 1".
2. Criar o "Post-it" (*Issue*)

Em vez de um simples cartão, o ideal é criar uma *Issue*, pois permite comentários e *checklists* interativas:
   - No quadro, cliquem em "+ Add item" na coluna "Todo".
   - Escrevam um título (p/ex.: "Objetivo SMART").
   - Cliquem no "+" e selecionem "Create new issue".
   - Cliquem no "Blank Issue".
   - Podem alterar o título do "Issue" e escrever uma descrição (p/ex., *Checklist* de autocorreção). Para a tarefa de definição do Objetivo do Projeto, podem colar o seguinte código *Markdown*:

```
### 🎯 Objetivo SMART
> [Escrever aqui a frase final do objetivo]

### ✅ Auto-Correção (Checklist)
- [ ] **S (Específico):** O que vamos prever/analisar está claro?
- [ ] **M (Mensurável):** Definimos uma métrica?
- [ ] **A (Atingível):** O nosso dataset permite isto?
- [ ] **R (Relevante):** Este objetivo é central para o tema?
- [ ] **T (Temporal):** Está associado a um Milestone/Data?
```



Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: presidencia@iscac.pt; www.iscac.pt