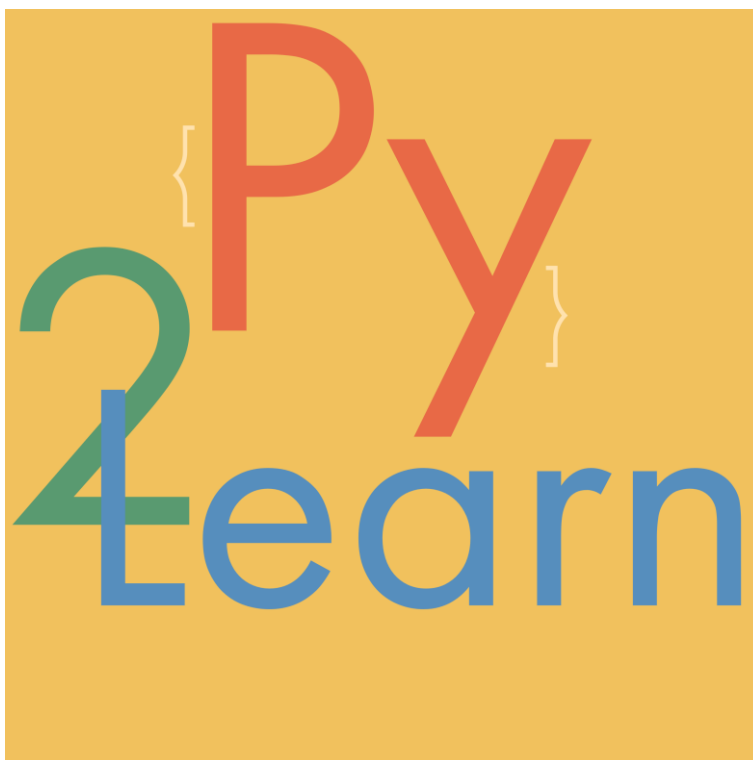


ADEETC – Área Departamental de Engenharia de Eletrónica e
Telecomunicações e de Computadores

LEIM - Licenciatura em Engenharia Informática e Multimédia

Unidade Curricular de Projeto



Aluna:

Carla Varela nº 40962

Orientador:

Prof. Dr. João Beleza

Setembro de 2019

Resumo

Neste projeto desenvolveu-se um curso da linguagem de programação Python usando a aplicação Web Jupyter Notebook, aplicação esta que permite ao aluno não só consultar os conteúdos dos diversos módulos existentes, no seu browser, como também, alterar e executar as linhas de código disponíveis, de forma a que este possa aprender de uma forma mais interativa.

Agradecimentos

Em primeiro lugar agradeço a Deus por ter-me dado forças para nunca desistir e ter ido até ao fim, de seguida agradeço aos meus pais por terem acreditado em mim e pelos muitos amigos que de forma positiva me influenciaram nesta caminhada da vida.

Por fim, agradeço aos professores pela enorme paciência e dedicação para comigo, apesar de muitos percalços, me motivara a não desistir e a dar o melhor de mim.

A todos os professores que durante todo o meu percurso académico foram
pacientes e procuraram passar aquilo que de melhor sabem,
principalmente ao meu orientador prof. Dr. João Beleza,
um especial agradecimento por ter acreditado no meu potencial.

Muito obrigado a todos.

Índice

Resumo	iii
Agradecimentos.....	v
Lista de Figuras.....	x
1 Introdução.....	1
2 Trabalho Relacionado	2
3 Modelo Proposto	3
3.1 Requisitos	4
3.1.1 Caracterização do Projeto.....	4
3.1.2 Casos de Utilização.....	5
3.1.3 Diagrama de Atividades	6
3.2 Fundamentos.....	7
3.2.1 Informações base dos notebooks	7
3.2.2 Tipo de notebooks entre outras informações	7
3.2.3 Ferramenta de Controle de Versões	7
3.2.4 Algumas funcionalidades dos notebooks.....	8
Abordagem.....	10
4 Implementação do Modelo	12
4.1 Funcionamento do <i>Jupyter</i>	12
4.2 Obter Conteúdos e Criar os Notebooks.....	13
4.3 Disponibilizar os Notebooks Online.....	14
4.4 Criar os Testes dos Módulos	19
4.5 Material Desenvolvido.....	21
5 Validação e Testes	22
6 Conclusões e Trabalho Futuro	26
A. Apêndice A.....	27
7 Bibliografia	28

Lista de Figuras

Figura 1 - Diagrama dos Casos de Utilização	6
Figura 2 - Diagrama de Atividades do Projeto	6
Figura 3 - Correr Célula.....	8
Figura 4 - Criar Célula.....	8
Figura 5 - Tipo de Célula.....	8
Figura 6 - Apagar Célula	9
Figura 7 - Guardar Notebook	9
Figura 8 - Baixar Notebook	9
Figura 9 - Jupyter Book.....	10
Figura 10 - Navegador do Anaconda.....	12
Figura 11 - Diretoria do Jupyter	13
Figura 12 - Criar Notebook.....	14
Figura 13 - Modelo de Funcionamento do Curso	14
Figura 14 - Inserir link do GitHub no Binder	15
Figura 15 - Executar Repositório no Binder	15
Figura 16 - Copiar Link do Binder	16
Figura 17 - Adicionar Link no README.md	16
Figura 18 - Criar Página GitHub	17
Figura 19 - Obter Link da Página do GitHub	17
Figura 20 - Página do GitHub Criada	18
Figura 21 - Design do Teste.....	20

1 Introdução

Neste projeto desenvolveu-se um curso da linguagem de programação Python usando a aplicação Web Jupyter Notebook. Tratando-se de uma aplicação Web, o curso pode ser disponibilizado online. Tirando partido desta possibilidade o curso desenvolvido neste projeto está efetivamente disponível online.

Como já foi previamente dito, este projeto consiste em um Curso Online de Python desenvolvido em *notebooks* da *Jupyter*.

Jupyter é o nome de um projeto vasto cujo objetivo geral é desenvolver software aberto (open-source), standards abertos e serviços para computação interativa em diversas linguagens de programação, entre as quais Java, Javascript, C++, HTML, PHP, Markdown, etc. (Jupyter s.d.)

O projeto Jupyter desenvolve o Jupyter Notebook que é uma aplicação Web que permite criar e partilhar documentos que contêm texto, código executável a pedido, equações e visualizações. Trata-se da possibilidade de disponibilizar texto e programas de computador executáveis a pedido, num mesmo documento acessível a partir de um browser, um Notebook Jupyter. É um mecanismo semelhante aquele que foi originalmente desenvolvido para a linguagem de programação Wolfram Mathematica ([fala aqui uma referência](#)).

Este curso abrange diferentes temas divididos em vários módulos. Cada módulo corresponde a um documento Jupyter Notebook.

O acesso ao notebook de cada módulo faz-se através de um notebook principal, este disponibiliza também testes de avaliação que permitem ao aluno averiguar o progresso da sua aprendizagem.

O que motivou a desenvolver este projeto foi o facto de que seria algo muito útil principalmente para os alunos que estejam a tirar um curso (ou algo do género) e que exija que aprendam esta linguagem de programação para terem um melhor aproveitamento.

O facto de permitir aprender mexendo, tornou-se assim sendo a principal motivação, visto que, se muitos alunos tivessem ao seu dispor este curso, com certeza teriam aprendido muito mais rápido esta linguagem (como seria no meu caso).

2 Trabalho Relacionado

Atualmente existem vários trabalhos desenvolvidos em notebooks Jupyter. Mas na sua grande maioria estão centrados na análise e visualização de dados.

Relativamente a cursos de Python existem na Internet bastantes. Alguns dos exemplos mais conhecidos são:

- W3schools - <https://www.w3schools.com/python/> - possui conteúdos de Python, mas os programas não podem ser alterados, apesar de ser possível realizar 51 exercícios os quais precisamos completar com código de forma a sabermos o nosso progresso na aprendizagem. Os conteúdos não estão em português;
- Learn Python - <https://www.learnpython.org/en/Welcome> - não é exatamente um curso pois não dá ao utilizador o feedback da sua aprendizagem, mas, no entanto, é possível executar programas. Outra desvantagem é que não há muitos conteúdos em português;
- Udemy - <https://www.udemy.com/topic/python/> - o curso é totalmente pago;
- Datacamp - <https://www.datacamp.com/courses/intro-to-python-for-data-science> - os conteúdos gratuitos são muito limitados, para além de serem em inglês.

No entanto na maior parte dos cursos de Python disponíveis não existe a possibilidade de executar o código Python apresentado no curso diretamente no documento onde é apresentado, ou seja, não são interativos.

É aqui que reside a inovação proposta neste projeto:

- Disponibilizar um curso de Python interativo a partir de um browser.

3 Modelo Proposto

Aquilo que inicialmente foi proposto neste projeto é que se montasse um curso de Python na plataforma *Jupyter* que poderia respeitar as seguintes fases:

1. Definição das diretivas de apresentação dos conteúdos (o código está em células e é só executar, ou está em texto e tem que se fazer *copy/paste* para células ou não é apresentado e o formando tem que o produzir).
2. Desenvolvimento de *notebooks* do curso de forma estruturada e que permita a adição de partes adicionais no futuro. Definir como é que se navega entre *notebooks*.
3. Desenvolvimento de exercícios de correção automática.
4. Especificação da infraestrutura de alojamento do curso.

Relativamente à fase 1. o que ficou decidido fazer foi que os códigos disponibilizados dentro de cada módulo pudessem ser executados pelo utilizador.

A parte de desenvolvimento de *notebooks* de forma estruturada foi respeitada, mas só poderá ser realizada por alguém que tenha uma noção mais aprofundada de Python de forma a alterar o código que se encontra escondido do utilizador de forma a facilitar a visibilidade.

Os exercícios desenvolvidos em cada módulo vão enviando para o utilizador o feedback, à medida que vão sendo resolvidos, ou seja irá ser transmitido ao utilizador se o exercício resolvido está certo ou errado.

Todo o conteúdo do curso encontra-se alojado no *[GitHub](#) e será através da aplicação web *[Binder](#) que o utilizador terá acesso ao curso em si.

3.1 Requisitos

3.1.1 Caracterização do Projeto

3.1.1.1 *Objetivos*

O objetivo deste projeto é desenvolver um curso online de python com notebooks *Jupyter* divididos por tema, notebooks esses que permitirão ao utilizador não só executar o código Python, mas também testar cada um dos temas. O utilizador não precisa fazer autenticação ou registo.

3.1.1.2 *Público-Alvo do Projeto*

Qualquer pessoa com interesse em aprender a linguagem de programação Python.

3.1.1.3 *Metas a Alcançar*

Aquilo que se pretende no final da implementação deste projeto é que todas as fases propostas venham a ser alcançadas com sucesso. Caso seja possível há funcionalidades que poderão ser adicionadas como por exemplo, o suporte para outras línguas para além do português, nomeadamente o inglês.

3.1.1.4 *Funções do Sistema*

O sistema tem de:

- Permitir ao utilizador escolher o módulo que pretende aprender;
- Apresentar os conteúdos de cada módulo;
- Permitir ao utilizador realizar os testes;
- Dar feedback dos exercícios que o utilizador vai resolvendo;
- Permitir que o utilizador execute os programas apresentados em cada módulo proporcionando uma aprendizagem interativa;
- Esconder o código Python usado na construção os notebooks para que não se confunda com os exemplos de código Python usados no curso.

3.1.1.5 Atributos do Sistema

O sistema tem que ser:

- Simples de utilizar;
- Interativo;

3.1.2 Casos de Utilização

O diagrama na figura abaixo ilustra os 5 casos de utilização existentes neste projeto, que serão executadas por 2 agentes que o Aluno e o sistema. Designaremos o sistema por *Notebooks Jupyter*.

Cada um dos casos é caracterizado por:

- Esconder Funções – impede que o aluno tenha acesso ao código que se encontra por detrás da funcionalidade do curso em si, de forma a melhorar a sua visão e perceção dos conteúdos disponibilizados;
- Aceder Módulo – Para facilitar a navegação entre os diferentes módulos/notebooks, foi criado um menu **dropdown* na barra lateral dos notebooks que permite ao utilizador escolher qual o módulo que ele pretende aceder;
- Correr Células/ Executar Código – não foi criada exatamente uma função para esta funcionalidade visto que o utilizador tem que copiar os exemplos de código do conteúdo disponibilizado para o seu interpretador Python, ou criar uma célula do notebook para o executar;
- Realizar Testes – função que disponibiliza pelo menos 4 exercícios referentes a cada dois temas e que há medida que o utilizador vai acertando, vai acrescentando 1 ponto à sua pontuação;
- Enviar Feedback – No final de cada teste que o utilizador vai resolvendo, é-lhe fornecida a informação sobre o número de exercícios que realizou corretamente.

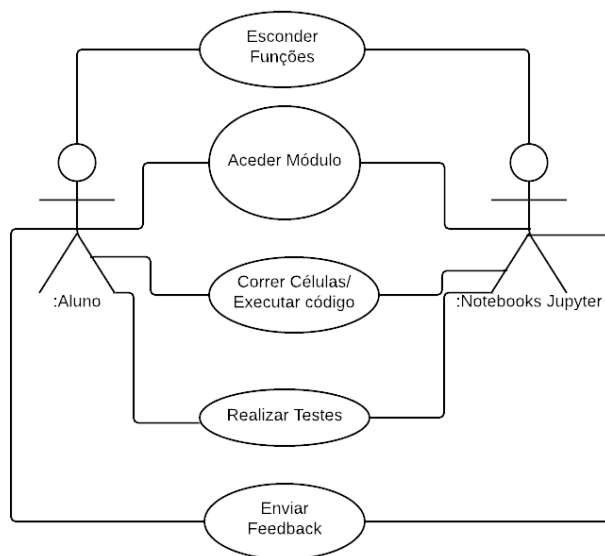


Figura 1 - Diagrama dos Casos de Utilização

3.1.3 Diagrama de Atividades

O diagrama de atividades correspondente a este projeto é o que se encontra na figura abaixo ilustrada. Pretende-se com este diagrama, ilustrar o funcionamento do curso.

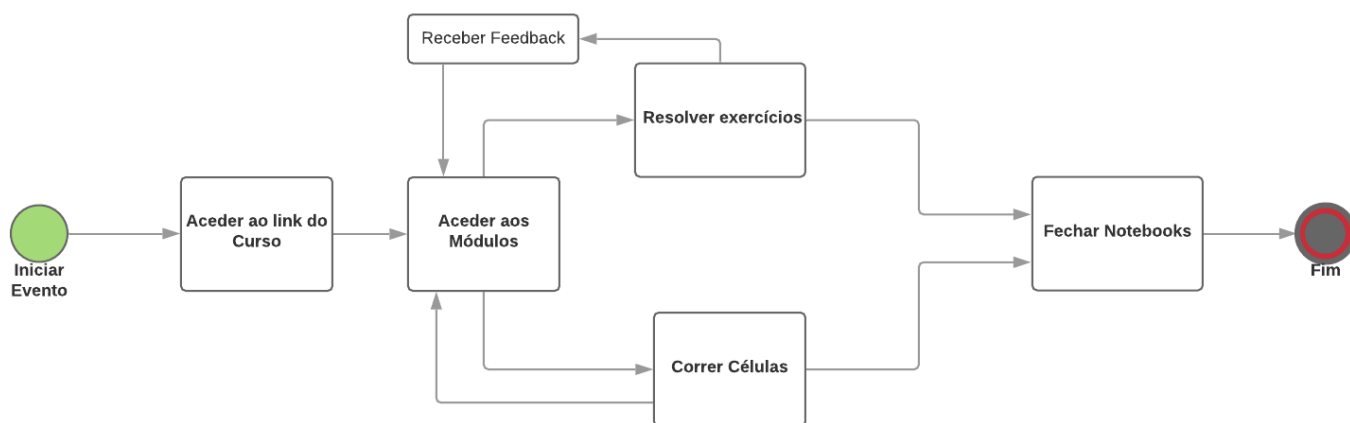


Figura 2 - Diagrama de Atividades do Projeto

3.2 Fundamentos

3.2.1 Informações base dos notebooks

Antes do desenvolvimento do projeto, houve a necessidade perceber o que é e como funcionam os *notebooks* da *Jupyter*.

Um notebook Jupyter é um documento JSON com um esquema que contém diversas células dispostas por ordem. As células podem adicionadas ou removidas.

Cada célula possui um tipo, sendo que elas podem ser classificadas como:

- Code - células de código executáveis que retornam ou não um resultado;
- Markdown – células de texto, recebe linguagem **Markdown* ou até mesmo **Html*;
- Heading – células de títulos;
- Raw NBConvert – células que serão descartadas quando o notebook for convertido para outros formatos. (Para já não será necessário aprofundar esta informação, uma vez que este tipo de célula não foi utilizado).

Os notebooks podem ser convertidos em documentos Html, slides, Latex, PDF, mas também podem ser obtidos (por download) como documentos *Jupyter Notebooks*, com a extensão “.ipynb”.

3.2.2 Tipo de notebooks entre outras informações

Este curso de Python foi desenvolvido utilizando notebooks do Python versão 3. A versão 2 embora esteja ainda disponível, já está em fim de vida.

O funcionamento dos notebooks da *Jupyter* depende da funcionalidade do *kernel* utilizado. O *kernel* é o sistema que executa os programas contidos nas células. Há kernels disponíveis para várias linguagens de programação, entre as quais Python 3. (GitHub s.d.)

3.2.3 Ferramenta de Controle de Versões

Usámos o GitHub, plataforma que usa sistemas open-source de controlo de versões Git, para efetuar o controlo de versões do desenvolvimento, partilhar o projeto e hospedar o curso na plataforma Binder tirando partido da possibilidade de hospedar notebooks Jupyter na plataforma Binder diretamente a partir do GitHub.

3.2.4 Algumas funcionalidades dos notebooks

Nesta secção mostram-se algumas funcionalidades dos notebooks Jupyter essenciais à utilização do curso desenvolvido.

- i. Para executar uma célula com código Python selecciona-se a célula em causa e clica-se no botão "Run" ou nas teclas "Shift" + "Enter".

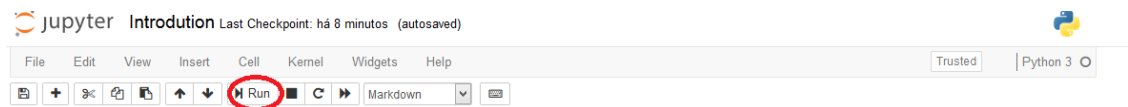


Figura 3 - Correr Célula

- ii. Para criar uma nova célula clica-se no botão "+".

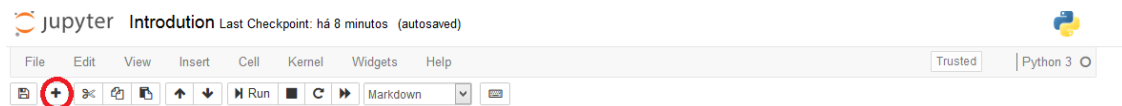


Figura 4 - Criar Célula

- iii. Visto que cada célula está associada a um tipo, para criar uma célula deve-se seleccionar o tipo pretendido no menu dropdown correspondente.



Figura 5 - Tipo de Célula

- i. Para remover uma célula, seleciona-se a célula em questão e em seguida “Delete Cells” na opção “Edit” do menu principal.

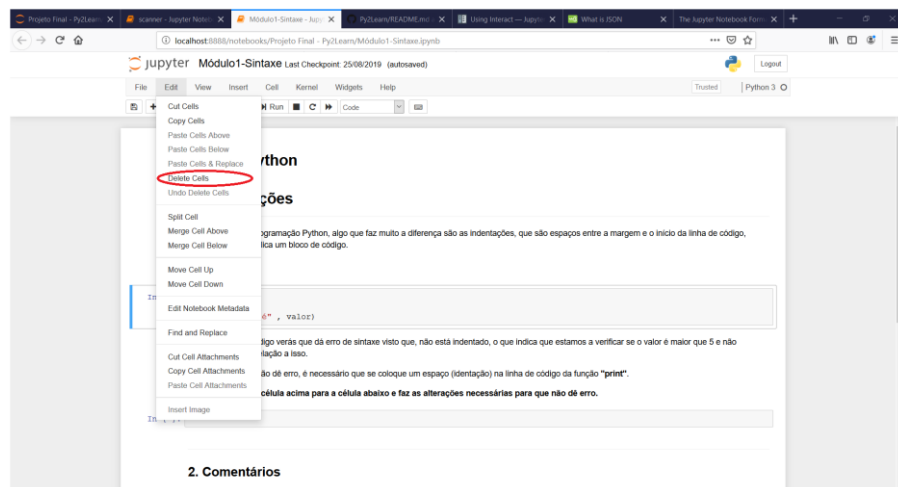


Figura 6 - Apagar Célula

- ii. Para guardar as alterações, clica-se no ícone do guardar da barra de ferramentas.

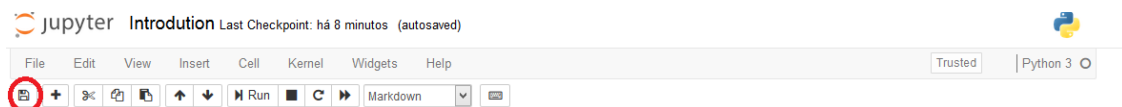


Figura 7 - Guardar Notebook

- iii. Para fazer download do notebook com as alterações que eventualmente possam ter sido feitas seleciona-se a opção "File", do menu principal, seguida da opção "Download", e, finalmente, "Notebook".

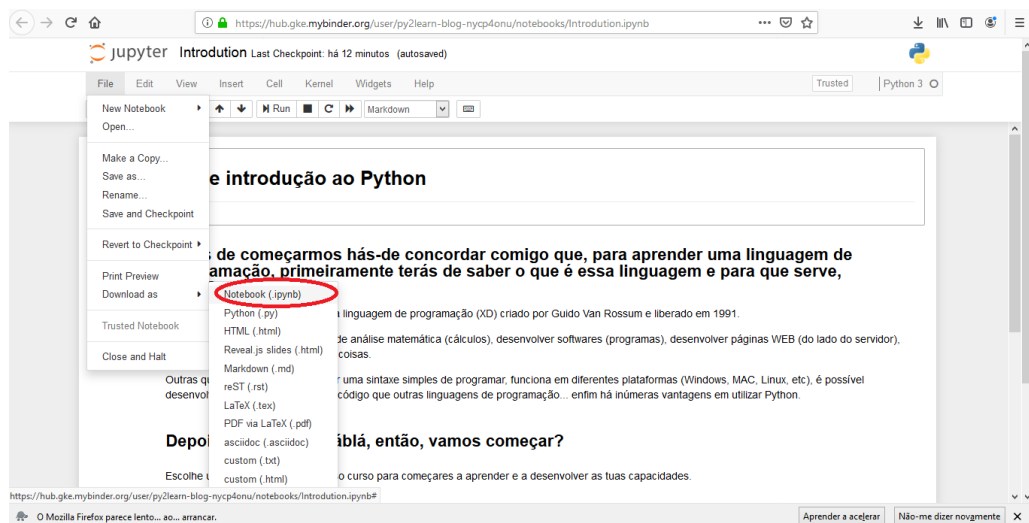


Figura 8 - Baixar Notebook

Abordagem

Os notebooks Jupyter não serão certamente a melhor forma de apresentar conteúdos a programadores principiantes ou a utilizadores não técnicos, na medida em que para além dos conteúdos propriamente ditos expõe também o código usado para construir esses conteúdos, permitindo assim que o utilizador altere inadvertidamente esse código obtendo resultados inadequados. Por outro lado, a presença do código de construção dos conteúdos dificulta a leitura e a navegação.

Estudou-se a possibilidade de resolver este problema usando Jupyter Book, uma extensão Jupyter Notebook para disponibilizar conteúdos suprimindo o código que os constrói. Esta solução não se revelou adequada essencialmente por duas razões. Por um lado, num curso Python não se pretende suprimir todo o código presente no notebook. Os exemplos de programação têm de permanecer visíveis. Esta possibilidade de suprimir apenas algumas células de código não é suportada pelo Jupyter Book. Por outro lado, a disponibilização de Jupyter Books através do GitHub não está atualmente funcional apesar de ser anunciada como tal.

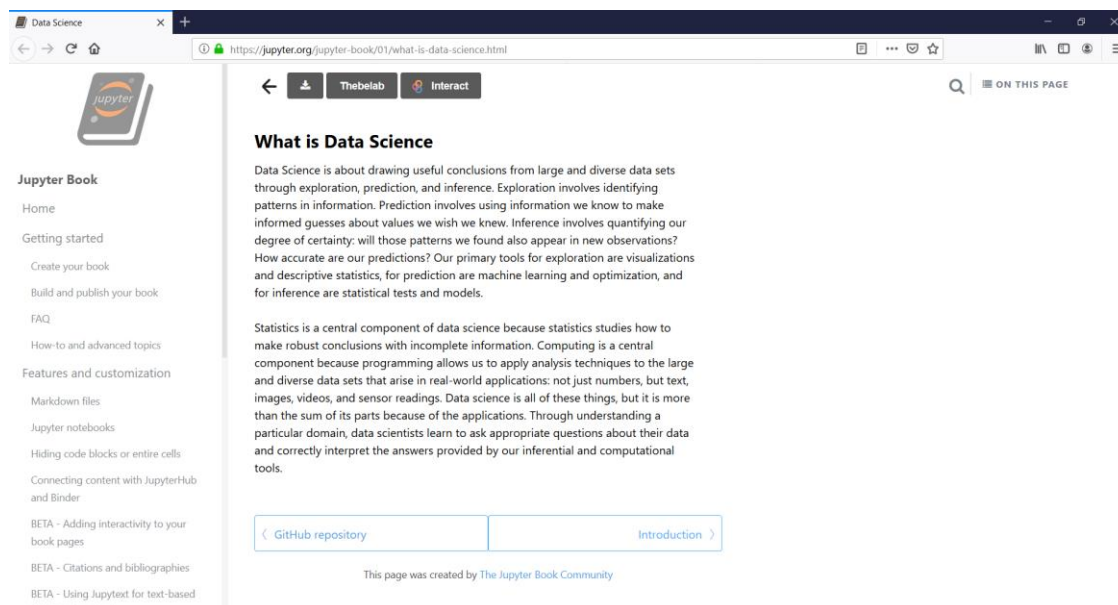


Figura 9 - Jupyter Book

Optou-se por fazer a hospedagem do curso recorrendo à plataforma Binder. Esta plataforma permite hospedar notebooks alojados em repositórios públicos do GitHub na sua versão interativa. Para além disso criar versões temporárias dos notebooks interativos para cada utilizador de modo a que as alterações de um utilizador num notebook não se façam sentir noutra utilizador que esteja a usar o mesmo notebook.

Para adicionar os testes, recorreu-se ao uso da linguagem HTML e JavaScript, apesar de que seria também possível criá-los através de Python, mas com estas linguagens seria possível alcançar um melhor aspeto da interface.

4 Implementação do Modelo

O desenvolvimento e a implementação do projeto envolveu os seguintes passos:

1. Estudar o funcionamento do *Jupyter Notebook*;
2. Obter conteúdos teóricos para os módulos, de acordo com os temas que se pretende abranger;
3. Criar notebooks para cada módulo com os conteúdos correspondentes;
4. Partilhar os meus notebooks online, sem que o original possa ser alterado pelos utilizadores e em que um utilizador possa fazer alterações no notebook de outro que esteja a consultar o curso em simultâneo;
5. Criar testes para os módulos;
6. Fazer testes finais;

4.1 Funcionamento do Jupyter

Relativamente ao Jupyter Notebook este passo encontra-se já detalhado no capítulo anterior.

Relativamente à obtenção do package Python Jupyter Notebook foi usada a distribuição Python Anaconda.

A distribuição [Anaconda](#) está disponível online e a sua instalação faz-se por download e execução da respetiva aplicação. Após ser finalizada a instalação, devemos abrir o navegador da Anaconda e clicar no “Launch” do “Notebook”, como poderá ver na imagem abaixo.

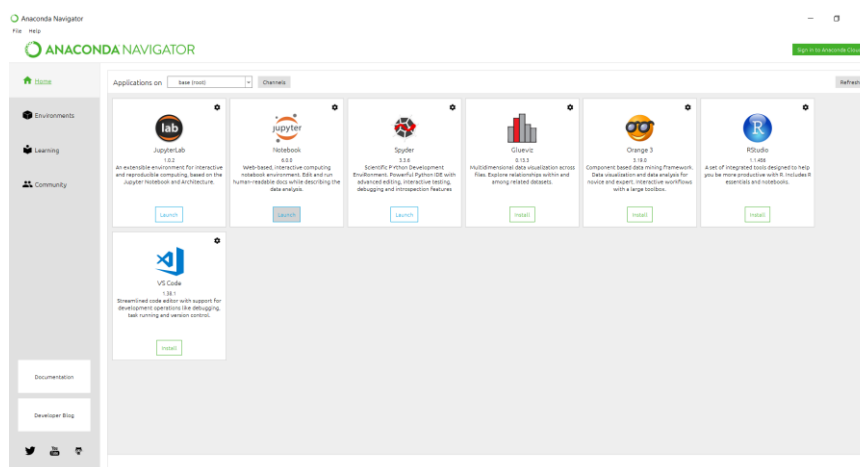


Figura 10 - Navegador do Anaconda

4.2 Obter Conteúdos e Criar os Notebooks

Os temas e os seus conteúdos foram pesquisados em diversos sites que especificam os assuntos relacionados a esta linguagem de programação, tais como [w3schools](#), [wikipedia](#), [caelum](#), entre outros.

Seguindo os passos da instalação do Anaconda até fazer “Launch”, abrir-se-à no nosso browser a diretoria principal do *Jupyter* no computador.

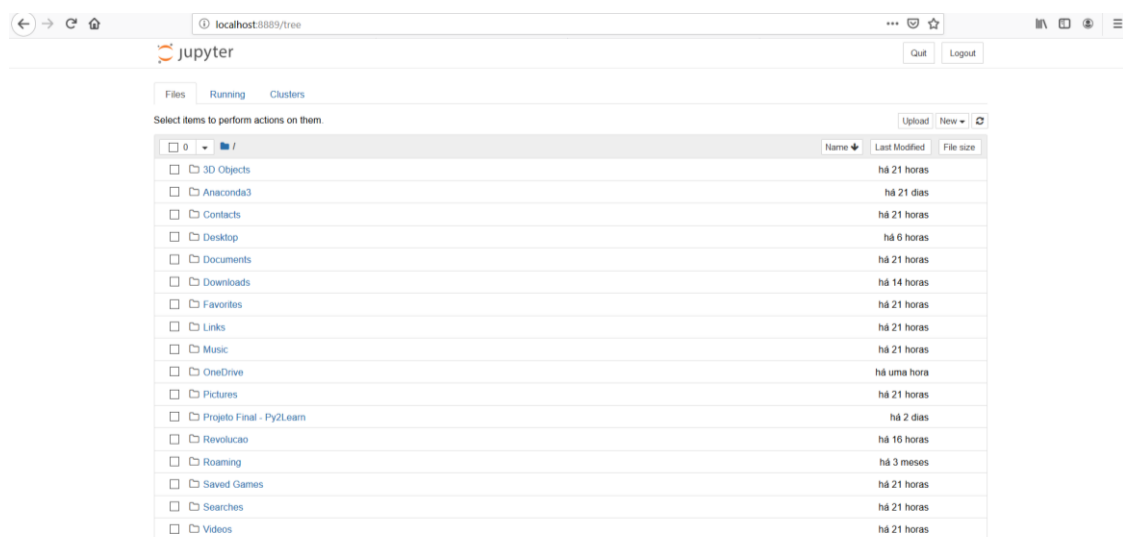


Figura 11 - Diretoria do Jupyter

Os notebooks a desenvolver deverão estar na diretoria principal do Jupyter Notebook.

Os outros tipos de ficheiros, como CSS e JavaScript (para os testes e outros aspetos visuais) podem ser criados como sendo ficheiros File Txt e depois alterar o nome (e a sua extensão) para o tipo de ficheiro pretendido.

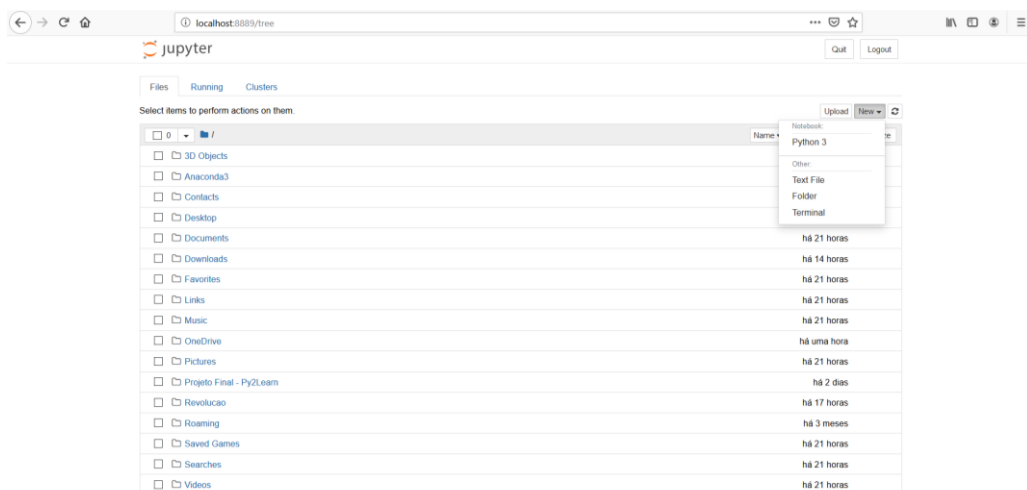


Figura 12 - Criar Notebook

4.3 Disponibilizar os Notebooks Online

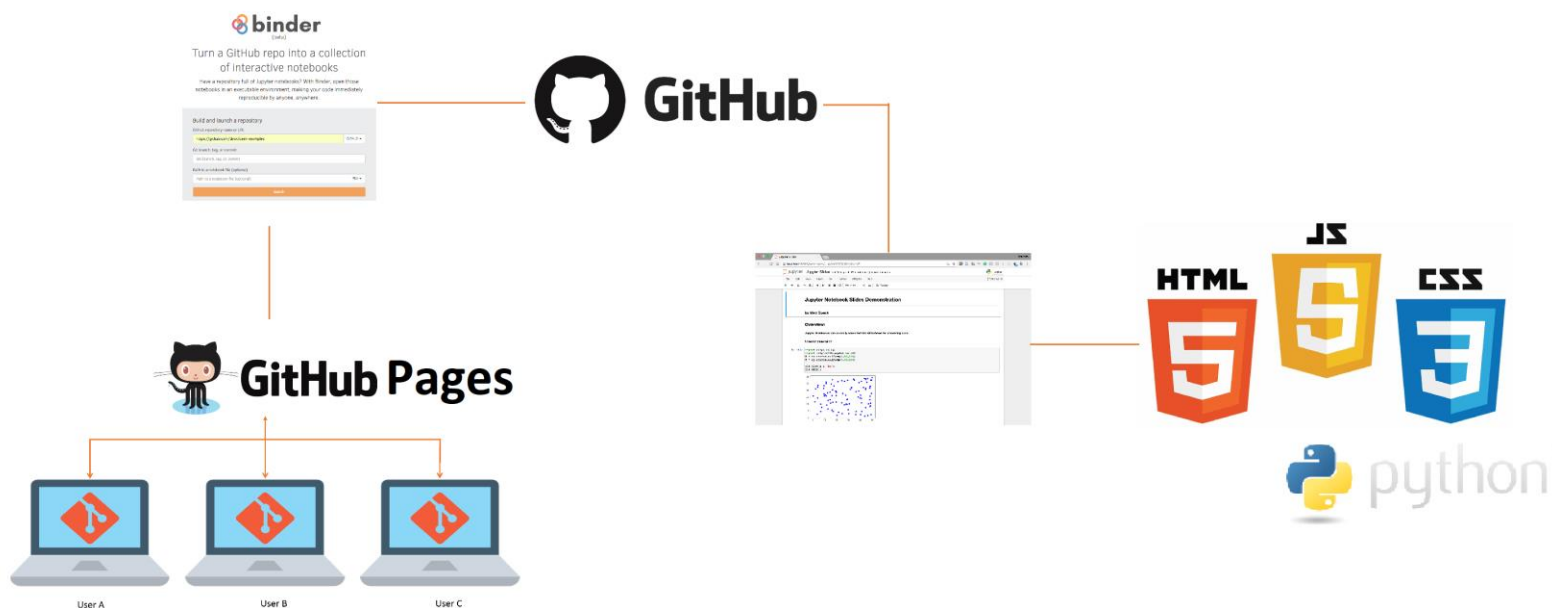


Figura 13 - Modelo de Funcionamento do Curso

O modelo escolhido para desenvolver este projeto está esquematizado no modelo acima.

O que basicamente este modelo quer dizer é que os utilizadores/alunos terão acesso ao link da página do GitHub, através dessa página ele será redirecionado para o repositório da GitHub onde se encontram os notebooks do curso.

Os passos para criar esse link de acesso são os seguintes:

1. Copiar o link do repositório do GitHub e adicionar na página do Binder;

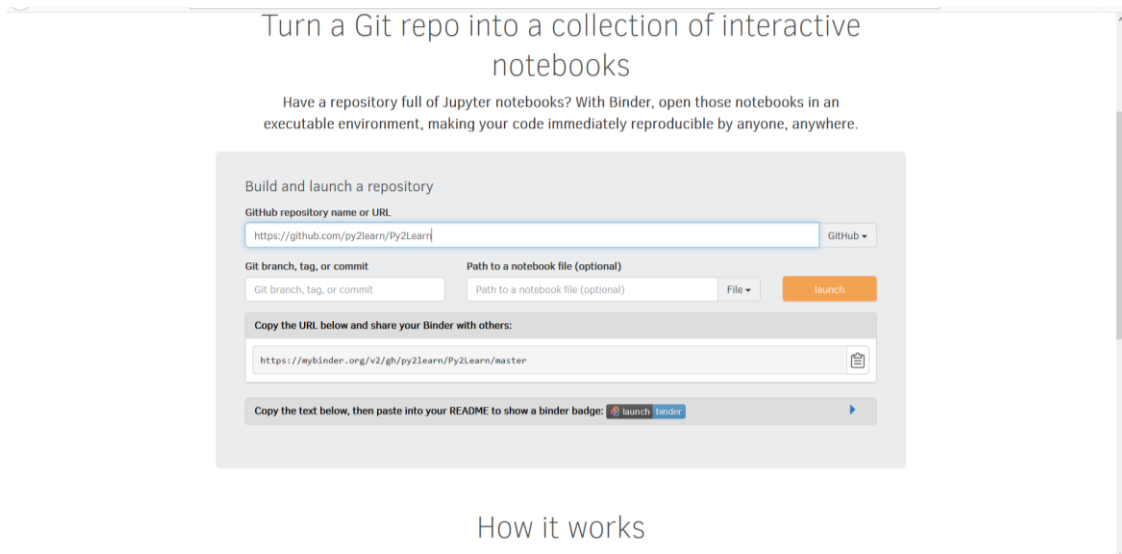


Figura 14 - Inserir link do GitHub no Binder

2. Clicar no botão “Launch”

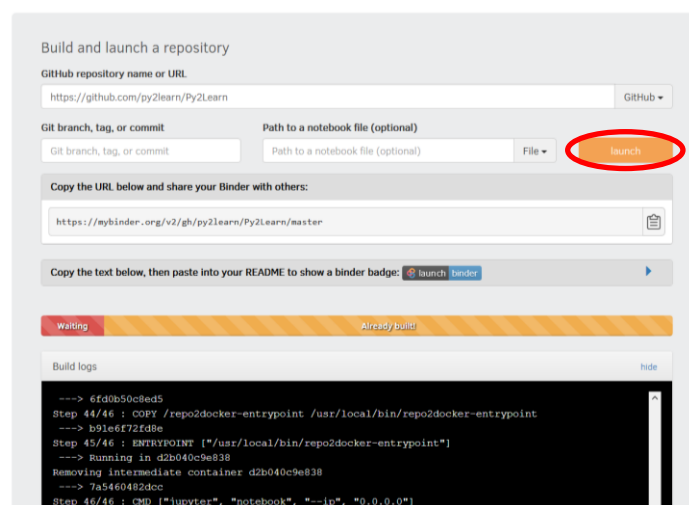


Figura 15 - Executar Repositório no Binder

3. Copiar o link fornecido pelo Binder para colocar na Página do GitHub de modo a que o utilizador tenha acesso aos notebooks;

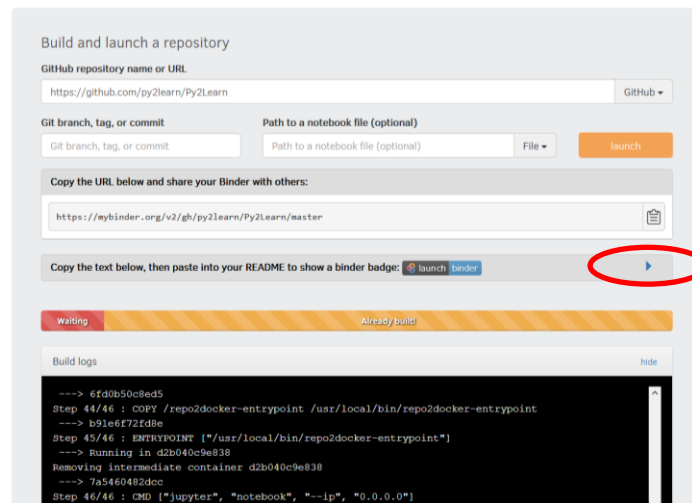


Figura 16 - Copiar Link do Binder

4. Adicionar o link ao ficheiro README.md para que esteja na página inicial do GitHub Page.

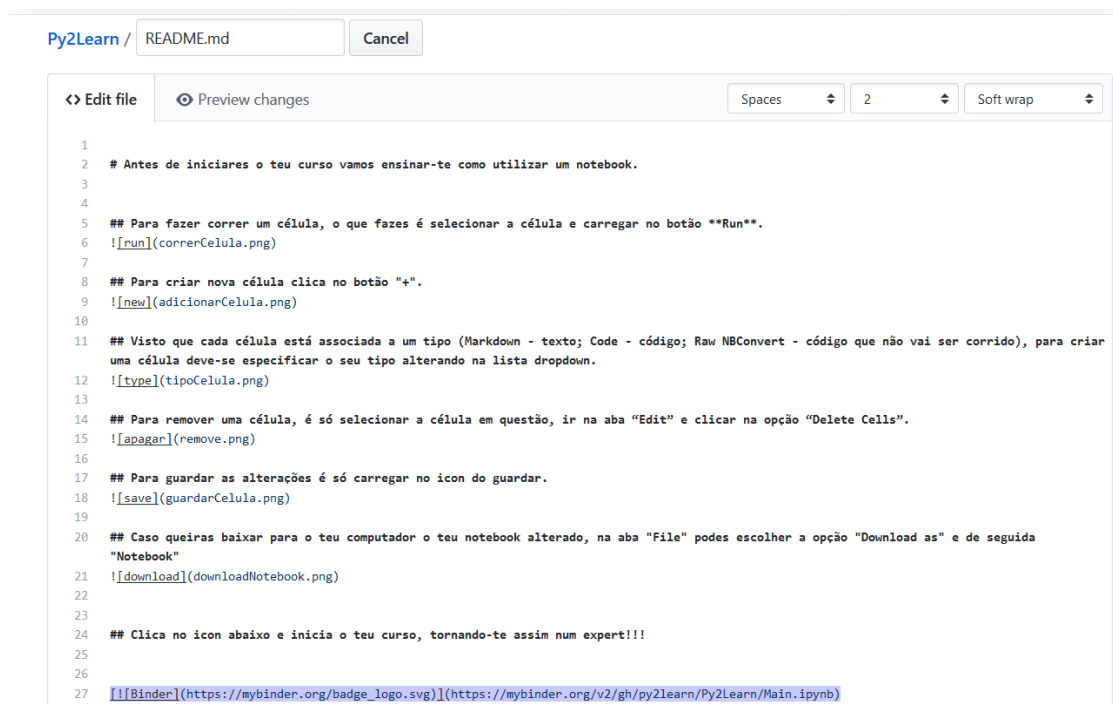


Figura 17 - Adicionar Link no README.md

5. Gerar uma GitHub Page indo ao “Settings”, na aba “GitHub Page” e copiar o link do mesmo. Esse é o link que pode ser compartilhado para os utilizadores.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <https://py2learn.github.io/Py2Learn/>

Source
Your GitHub Pages site is currently being built from the `master` branch. [Learn more.](#)

master branch ▾

Select source

✓ master branch
Use the `master` branch for GitHub Pages.

master branch /docs folder
Use only the `/docs` folder for GitHub Pages.

None
Disable GitHub Pages.

Save

☒ **Enforce HTTPS**
— Required for your site because you are using the default domain (`py2learn.github.io`)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

Figura 18 - Criar Página GitHub

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <https://py2learn.github.io/Py2Learn/>

Source
Your GitHub Pages site is currently being built from the `master` branch. [Learn more.](#)

master branch ▾

Theme Chooser
Select a theme to publish your site with a Jekyll theme. [Learn more.](#)
Your site is currently using the Cayman theme.

Change theme

Custom domain
Custom domains allow you to serve your site from a domain other than `py2learn.github.io`. [Learn more.](#)

Save

☒ **Enforce HTTPS**
— Required for your site because you are using the default domain (`py2learn.github.io`)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

Figura 19 - Obter Link da Página do GitHub



Figura 20 - Página do GitHub Criada

4.4 Criar os Testes dos Módulos

Para criar os testes dos módulos, como já foi dito anteriormente procedeu-se à utilização da linguagem JavaScript.

Foi criado 1 teste por cada 2 Módulos, o que faz um total de 4 testes, visto que são 8 módulos, ou seja, o utilizador será avaliado de acordo com os conteúdos dos dois módulos anteriores, mas no entanto não é obrigatório que o aluno leia os módulos antes de fazer os testes. Os testes encontram-se disponibilizados na lista de conteúdos (menu de navegação) no fim de cada 2 módulos.

Para utilizar código JavaScript nos notebooks, há a necessidade de importar o método HTML, visto que é o HTML que será lido neste contexto e que irá executar a linguagem JavaScript embutida em si, e isso se faz com o seguinte código:

```
from IPython.core.display import display, HTML
```

Depois de importar, iremos utilizar este método para poder colocar o código html/javascript pretendido (utilizando as *tags dos mesmos).

Exemplo:

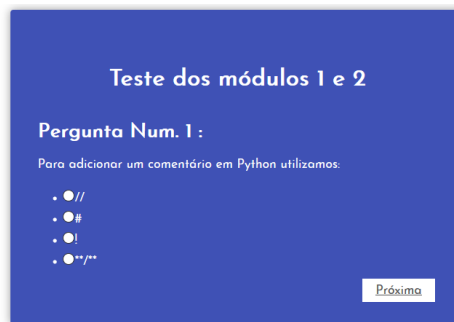
```
HTML( '''<html><javascript>'''código  
javascript'''</javascript></html>''' )
```

Para as perguntas, foram criados ficheiros javascript que depois são referenciadas no notebook referente a cada teste. No ficheiro javascript elas estão colocadas dentro de uma variável do tipo **array* chamada de *allQuestions* em que cada pergunta corresponde a uma pergunta, variável “question”, um conjunto de 4 respostas possíveis, variável “options” e um valor correspondente do índice da resposta correta, variável com o nome “answer”.

Basicamente o que a função que se encontra dentro do ficheiro faz é para além de disponibilizar os links de anterior e próxima pergunta (escondendo ou mostrando de acordo com a necessidade), mostrar as perguntas e as suas opções de escolha, o que faz também é somar os pontos à medida que o utilizador vai acertando e depois mostrar o resultado final.

Nos ficheiros de teste, dentro do código HTML que é chamado, também se faz referência aos ficheiros CSS utilizados de forma a dar um melhor aspeto ao teste e de forma a dispor todos os elementos no devido lugar. Na Figura 21 é possível ver o layout do teste criado.

Através do link <https://py2learn.github.io/Py2Learn/> é possível aceder ao Curso. Ao entrar no repositório do Curso, deve escolher o notebook “Main” e daí terá acesso a tudo o que precisa.



O código deste notebook está oculto de forma a facilitar a leitura e interação, para poder aceder ao mesmo clique [aqui](#).

Figura 21 - Design do Teste

4.5 Material Desenvolvido

Para ser mais simplificada a navegação do utilizador pelos notebooks foram criados os ficheiros “Main.ipynb” e “Introduction.ipynb” que dão uma breve introdução do Curso e da linguagem de programação Python, respetivamente.

Os temas da linguagem Python abordados neste curso encontram-se abaixo com os seus respetivos ficheiros Notebooks.

- 1) Sintaxe – “Sintaxe.ipynb”;
- 2) Variáveis
 - 2.1. Booleanos – “Variaveis.ipynb”;
 - 2.2. Números – “Numeros.ipynb”;
 - 2.3. Strings – “Strings.ipynb”;
- 3) Coleções
 - 3.1. Listas – “Colecoes.ipynb”;
 - 3.2. Tuplos – “Tuplos.ipynb”;
 - 3.3. Conjuntos – “Conjuntos.ipynb”;
 - 3.4. Dicionários – “Dicionarios.ipynb”;
- 4) Operadores – “Operadores.ipynb”;
- 5) Controlo de Fluxo – “Fluxo.ipynb”;
- 6) Funções – “Funcoes.ipynb”;
- 7) Classes/ Objetos – “ClassObjetos.ipynb”;
- 8) Heranças – “Herancas.ipynb”.

Testes Desenvolvidos e seus respetivos ficheiros:

- Teste 1 – “teste1.ipynb” – Testes referentes aos temas 1 e 2;
- Teste 2 – “teste2.ipynb” – Testes referentes aos temas 3 e 4;
- Teste 3 – “teste3.ipynb” – Testes referentes aos temas 5 e 6;
- Teste 4 – “teste4.ipynb” – Testes referentes aos temas 7 e 8;

Os ficheiros “quiz.js”, “quiz.js2”, “quiz.js3” e “quiz4.js”, são ficheiros *JavaScript*, respetivos de cada teste criado e abordado anteriormente.

O ficheiro “quiz.css” é o ficheiro que possui a descrição do estilo associado a todos os testes.

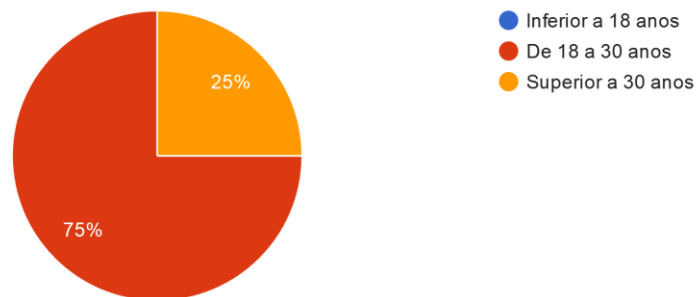
O ficheiro “README.md” é o ficheiro que contém as informações básicas que o utilizador precisa saber para fazer o curso.

5 Validação e Testes

Resumo do resultado de 4 testes feitos através de formulários da Google:

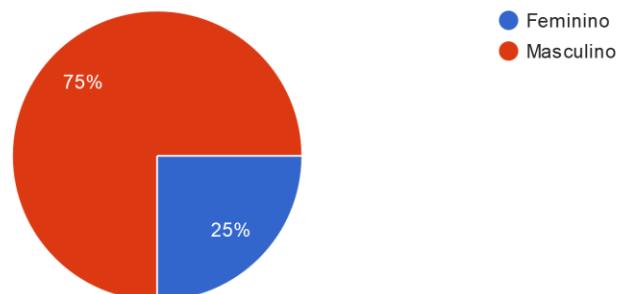
Idade

4 respostas



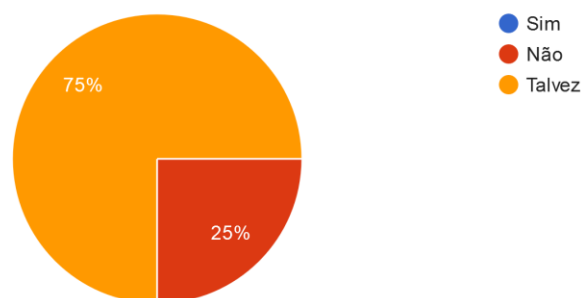
Sexo

4 respostas



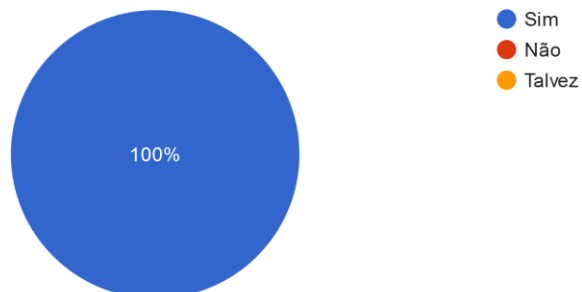
Foi fácil navegar entre os notebooks?

4 respostas



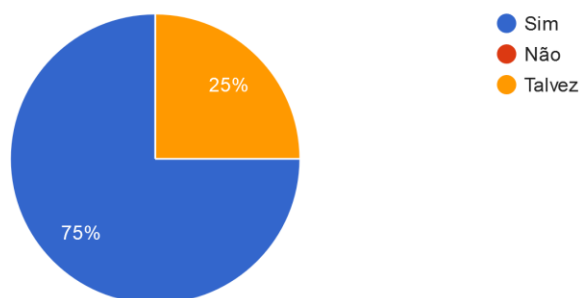
O curso é de fácil interacção?

4 respostas



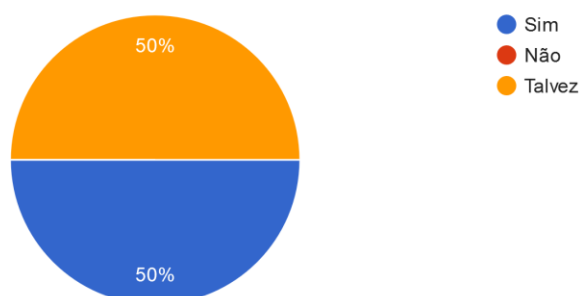
Foi fácil aprender os conteúdos?

4 respostas



O conteúdo satisfaz a sua necessidade?

4 respostas



Há alguma coisa que não gostou neste curso?

2 respostas

podia haver algum objectivo a alcançar

A organização

Alguma sugestão para melhorar o curso?

3 respostas

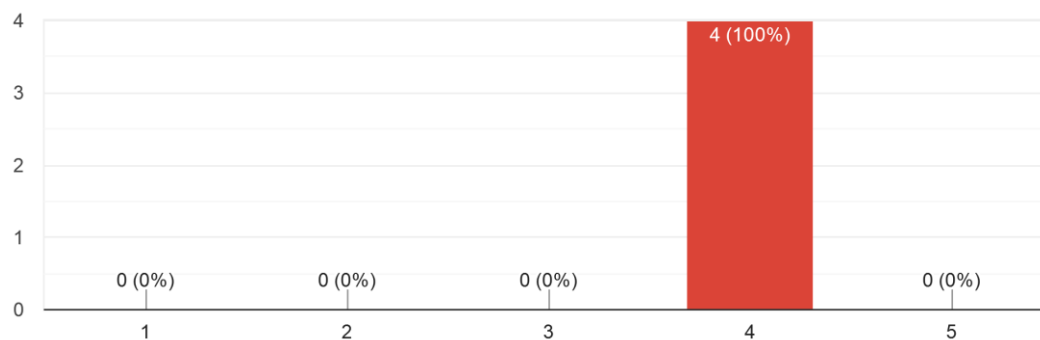
navegabilidade e proposta de sequência

Talvez colocar pontuações para os utilizadores sentirem mais motivados para os próximos níveis

Ter páginas de teste de conhecimento entre módulos

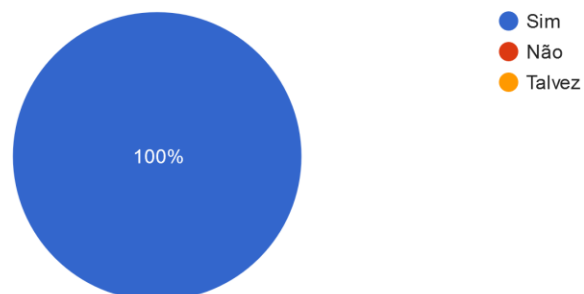
Acha este curso útil?

4 respostas



Recomendaria o curso a alguém?

4 respostas



Daquilo que foi proposto mudar, aquilo que foi melhorado foi a navegação entre notebooks, a realização de testes de forma a que o utilizador tenha noção do quanto aprendeu.

6 Conclusões e Trabalho Futuro

Este projeto eu creio que irá ajudar a muitos alunos, principalmente aos que estão a iniciar o curso, mas no entanto vejo que poderia melhorar algumas questões que não foram bem satisfeitas, tais como a interface disponibilizada para aceder aos notebooks do curso, visto que ninguém iria gostar de entrar num site/ aplicação e ver uma lista de coisas que não entende bem a sua finalidade logo diretamente, mas ainda assim foram alcançados os objetivos de permitir o acesso a várias pessoas, sem alterar o conteúdo do ficheiro inicialmente criado, criar testes que permitissem aos alunos receber um feedback de como se estão a sair, entre outros.

Os trabalhos para o futuro neste projeto seriam:

- Conteúdos noutras línguas;
- Testes diferentes para cada utilizador e que fossem aleatórios;
- Permitir com que o aluno faça registo e login de forma a que a cada aula das unidades curriculares onde aprende Python, à medida que vão dando os temas, os alunos vão resolvendo os problemas e enviar o feedback para o professor ou até mesmo para que eles possam guardar a sua evolução;
- Caso fosse possível realizar uma página com as pontuações dos alunos de forma a motivá-los a estudarem mais e mais e concorrerem entre si.

A. Apêndice A

Glossário

*GitHub- é uma plataforma online de hospedagem de código com controle de versão usando o Git que permite com que os programadores ou simples utilizadores registados publiquem projetos privados ou públicos de qualquer parte do mundo (GitHub s.d.);

*Binder- é uma aplicação web que permite transformar um repositório da GitHub numa coleção de notebooks interativos, ou seja, vais nos permitir utilizar o nosso repositório da GitHub para abrir os notebooks que lá existem sem haver a necessidade de instalar o *Jupyter* para executar os notebooks. Esta aplicação não só permite executar os notebooks, mas também criar o link do nosso repositório e partilhá-lo com quem nós quisermos (Binder s.d.);

* menu *dropdown* – Um menu dropdown ou menu suspenso é um elemento de interface com o não é similar a uma lista, que permite que o usuário escolha um valor de uma lista de opções que "cai para baixo". Quando o menu dropdown está inativo, ele esconde as opções do menu, economizando espaço na tela (Wikipédia, a enciclopédia livre s.d.);

*tags – são estruturas de linguagem de marcação contendo instruções, tendo uma marca de início e outra de fim para que o navegador possa renderizar uma página (Wikipédia, a enciclopédia livre s.d.) ;

*array – é uma estrutura de dados que armazena uma coleção de elementos (Wikipédia, A enciclopédia livre s.d.);

*HTML – é uma linguagem de marcação utilizada para criar páginas Web (w3schools s.d.);

*Markdown – linguagem que transforma o texto em conteúdo HTML válido (Wikipédia, A enciclopédia livre s.d.);

*documento JSON – formatos mais utilizados para a comunicação entre serviços web, baseados em programação JavaScript (w3schools s.d.).

7 Bibliografia

Binder. s.d. <https://mybinder.org/>.

Code Actually. s.d. <https://codeactually.com/interactivequiz.html>.

codecademy. s.d. <https://www.codecademy.com/learn/learn-python>.

datacamp. s.d. <https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook>.

Git. s.d. <https://git-scm.com/>.

GitHub. s.d. <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>.

IPython. s.d. <https://ipython.org/ipython-doc/dev/notebook/nbformat.html>.

Jupyter. s.d. <https://jupyter.org>.

Jupyter. s.d. <https://jupyter.org/jupyter-book/intro.html>.

Jupyter Notebook. s.d. <https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Importing%20Notebooks.html>.

Learn Python. s.d. <https://www.learnpython.org/>.

MDN web docs. s.d. <https://developer.mozilla.org/pt-PT/docs/Web/HTML>.

Quora. s.d. <https://www.quora.com/What-is-the-HTML-code-for-a-quiz>.

Sitepoint. s.d. <https://www.sitepoint.com/simple-javascript-quiz/>.

w3schools. s.d. https://www.w3schools.com/python/python_lists.asp.

w3schools. s.d. https://www.w3schools.com/whatis/whatis_json.asp.

w3schools. s.d. <https://www.w3schools.com/html/>.

Wikipédia, a enciclopédia livre. s.d. <https://pt.wikipedia.org/wiki/GitHub>.

Wikipédia, A enciclopédia livre. s.d. [https://pt.wikipedia.org/wiki/Arranjo_\(computa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Arranjo_(computa%C3%A7%C3%A3o)).

Wikipédia, A enciclopédia livre. s.d. [https://pt.wikipedia.org/wiki/Tag_\(linguagens_de_marca%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Tag_(linguagens_de_marca%C3%A7%C3%A3o)).

Wikipédia, A enciclopédia livre. s.d. https://pt.wikipedia.org/wiki/Padr%C3%A3o_aberto.

Wikipédia, A enciclopédia livre. s.d. https://pt.wikipedia.org/wiki/Menu_drop-down.

Wikipédia, A enciclopédia livre. s.d. https://pt.wikipedia.org/wiki/C%C3%B3digo_aberto.