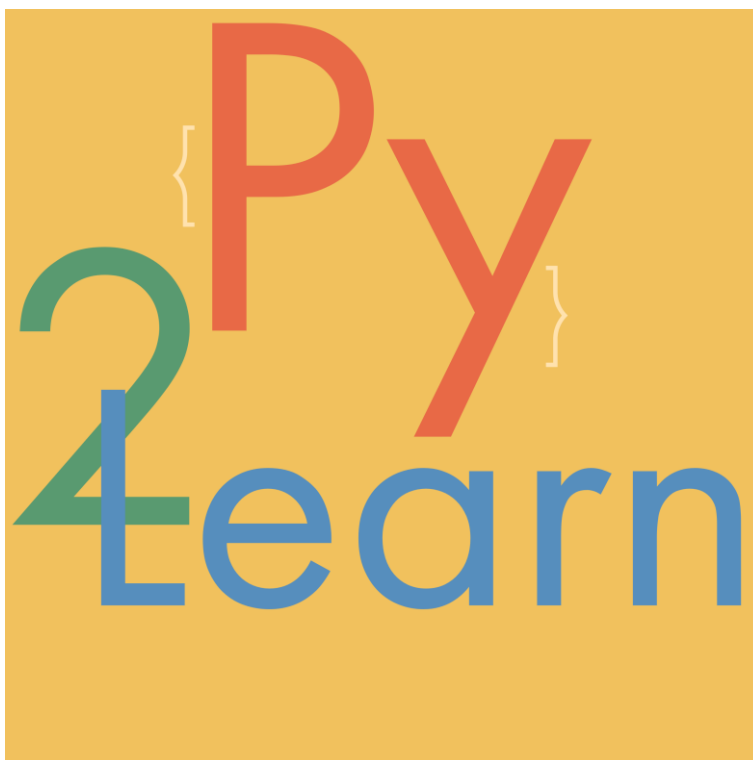


ADEETC – Área Departamental de Engenharia de Eletrónica e
Telecomunicações e de Computadores

LEIM - Licenciatura em Engenharia Informática e Multimédia

Unidade Curricular de Projeto



Aluna:

Carla Varela nº 40962

Orientador:

Dr. João Beleza

Resumo

Este projeto trata-se de um curso da linguagem de programação Python orientada a objetos (codecademy s.d.) desenvolvida através da interface *Jupyter*, interface esta que permite ao aprendiz não só aprender os conteúdos dos diversos módulos existentes, mas também alterar e executar as linhas de código disponíveis no mesmo, de forma a que este possa aprender de uma forma mais interativa.

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).

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, ainda que devido a muitos percalços motivaram-me 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 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 das notebooks	7
3.2.2 Tipo de notebooks entre outras informações	7
3.2.3 Ferramentas de Controle de Versões.....	8
3.2.4 Algumas funcionalidades dos notebooks.....	8
3.3 Abordagem	10
4 Implementação do Modelo	12
4.1.1 Funcionamento do <i>Jupyter</i>	12
4.1.2 Obter Conteúdos e Criar os Notebooks	13
4.1.3 Disponibilizar os Notebooks Online	14
4.1.4 Criar os Testes dos Módulos.....	19
5 Validação e Testes	21
6 Conclusões e Trabalho Futuro	25
A. Apêndice A.....	26
7 Bibliografia.....	27

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

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

Antes de tudo há necessidade de entender o que é o *Jupyter*. O que há para dizer é que é um projeto que existe basicamente para desenvolver através dos *notebooks softwares* (programas) *open-source* (de livre consulta, examinação ou modificação, ou seja sem necessidade de pagar licenças), *open-standards* (podem ser utilizados por qualquer pessoa) e serviços para computação interativa em dezenas de linguagens de programação, entre os quais Python, Java/ Javascript, C++, HTML, PHP, Markdown, entre outros. (Jupyter s.d.)

Obs: Nomeadamente os *notebooks* deste projeto foram escritos recorrendo à linguagem Python, em termos de conteúdos dos exemplos (códigos executáveis) e também à linguagem HTML e Markdown no que toca à parte dos conteúdos de texto. Neste projeto foram utilizados notebooks Python 3, visto que são os mais atuais.

Este curso possui vários módulos/ temas, sendo que cada um possui os seus conteúdos disponibilizados dentro de um *notebook* (aplicação web que contém códigos executáveis).

Os *notebooks* de cada módulo são acedidos através de um *notebook* principal que não só disponibiliza os conteúdos dos mesmos, mas também os testes de cada tema, testes estes que irão transmitir para o utilizador/ aluno o feedback da sua aprendizagem através da sua pontuação à medida que vai resolvendo os “problemas”.

2 Trabalho Relacionado

Atualmente existem vários trabalhos desenvolvidos em notebooks da *Jupyter*, só que mais voltados para análise de casos, trabalhos de estatística, ou seja, mais voltados para a área de análises matemáticas.

Existem muitos sites com conteúdos de Python por aí que podem ser executados, bem como:

- W3schools - <https://www.w3schools.com/python/> - possui conteúdos de Python, mas os códigos não podem ser alterados, apesar de ser possível realizar 51 exercícios no qual precisamos completar com código de forma a sabermos o quanto nós aprendemos da matéria. Os conteúdos não estão em português;
- Learn Python - <https://www.learnpython.org/en/Welcome> - só que não é exatamente um curso pois não dá ao utilizador o feedback da sua aprendizagem, mas no entanto é possível executar os códigos. 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.

Com toda esta informação, podemos considerar que ainda não exista um projeto semelhante.

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 fossem copiados e colados numa célula onde pudessem ser executados.

A parte de desenvolvimento de *notebooks* de forma estruturada foi respeitada, mas só poderá ser realizada pelo gestor, ou seja, 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 dos mesmos, 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 se encontra 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*

Os objetivos deste projeto é obter um curso online de python desenvolvido em notebooks *Jupyter* divididos por tema, notebooks esses que permitirão ao utilizador não só executar os seus códigos, mas também testar cada um dos temas. O utilizador não precisa fazer autenticação ou registo.

3.1.1.2 *Público-Alvo do Projeto*

Qualquer indivíduo que tenha o interesse de 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 a fases propostas venham a ser alcançadas com sucesso e caso dê tempo para adicionar outras funcionalidades adicionais tais como, permitir com que seja possível mudar a linguagem, a exemplo do 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;
- Esconder as funções de funcionamento do curso.

3.1.1.5 Atributos do Sistema

O sistema tem que ser...

- Simples de utilizar;
- Simples de executar os códigos;
- Interativo com o utilizador;

3.1.2 Casos de Utilização

Neste projeto, assim como podem ver na figura do diagrama abaixo apresentado, teremos 5 casos de utilização, ou seja 5 funções que serão executadas pelos 2 possíveis agentes que são o Aluno e o sistema, a que ficou com o nome de *Notebooks Jupyter*.

O que cada um dos casos faz é o seguinte:

- Esconder Funções – impede com que o aluno não 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 o que tem de fazer é copiar um código do conteúdo disponibilizado, ou até mesmo criar um numa das células do notebook e executá-lo;
- 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 – a cada teste que o utilizador vai resolvendo, no final lhe é enviado o feedback de que ele acertou um determinado número de exercícios.

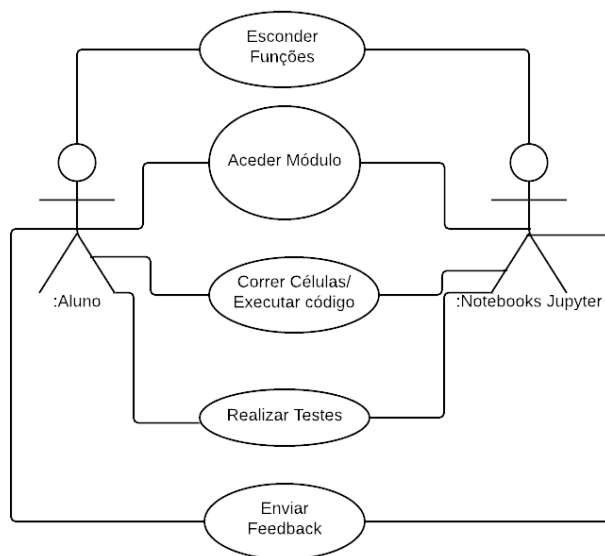


Figura 1 - Diagrama dos Casos de Utilização

3.1.3 Diagrama de Atividades

O diagrama de atividades correspondente a este projeto é o que se encontra na figura abaixo ilustrada. Desta forma será mais fácil de entender os passos de funcionamento do Curso.

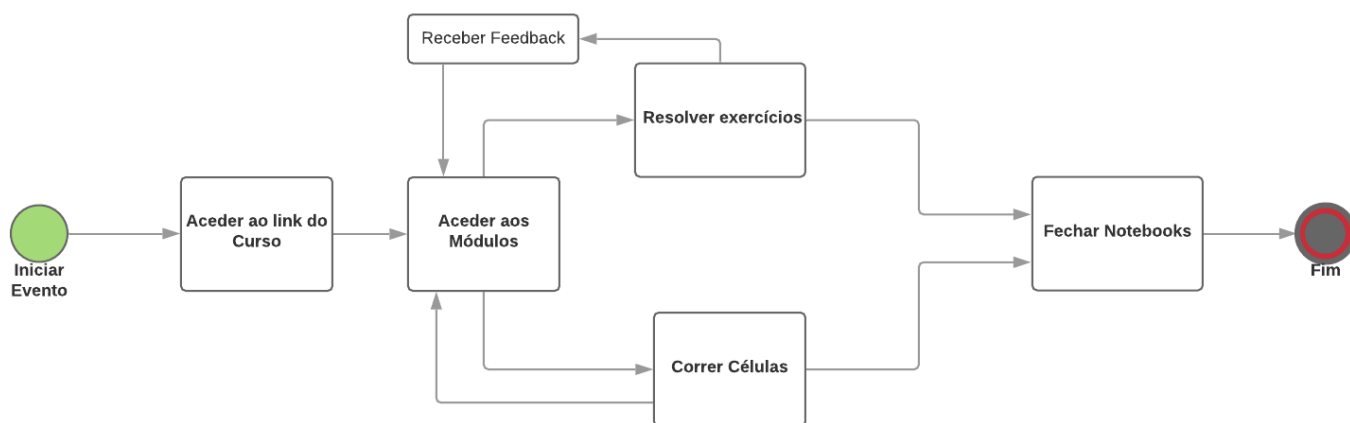


Figura 2 - Diagrama de Atividades do Projeto

3.2 Fundamentos

3.2.1 Informações base das notebooks

Antes de irmos para o desenvolvimento do projeto, há uma enorme necessidade de entender o que é e como funcionam os *notebooks* da *Jupyter*. Esse foi o primeiro passo dado para poder alcançar os principais objetivos deste projeto.

A plataforma *Jupyter* é basicamente um *documento JSON com um esquema que contém diversas células dispostas por ordem e podem ir sendo 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 em outros formatos. (Para já não será necessário aprofundar nesta informação, até porque este tipo de célula não foi utilizado).

Os notebooks podem ser convertidos como documentos Html, slides, Latex, PDF, mas também podem ser baixados como sendo 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 3, em vez do Python 2, visto que estes se encontram mais atualizados e que também são os mais recomendados pelo *Jupyter* devido às atualizações das funcionalidades e métodos.

O funcionamento dos notebooks da *Jupyter* dependem imenso da funcionalidade dos *kernels*, visto que estes são sistemas/programas que permitem com que os códigos sejam executáveis e com que seja possível utilizar outras linguagens de programação para o desenvolvimento das aplicações desejadas pelo seu gestor. (GitHub s.d.)

3.2.3 Ferramentas de Controle de Versões

Para um melhor controle de versões os *notebooks* encontram-se guardados na plataforma GitHub, plataforma esta que faz uso do *Git* que é um sistema *open-source* de controle de versões que facilita a gestão de conteúdos de projetos. A razão de escolha destes programas basicamente resume-se ao facto de facilitarem na partilha dos conteúdos.

3.2.4 Algumas funcionalidades dos notebooks

Antes de aprofundar ainda mais no que toca a desmistificar este projeto, iremos mostrar algumas coisas que podemos fazer nos notebooks:

- i. Para fazer correr uma célula, o que fazes é seleccionar a célula e carregar no botão “Run”.

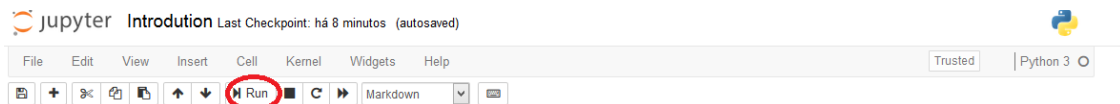


Figura 3 - Correr Célula

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

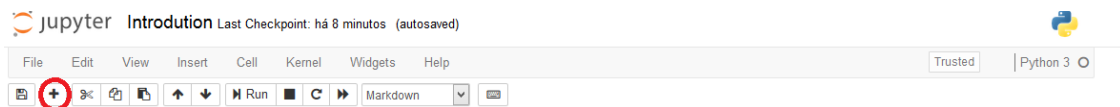


Figura 4 - Criar Célula

- iii. Visto que cada célula está associada a um tipo, para criar uma célula deve-se especificar o seu tipo alterando na lista dropdown.

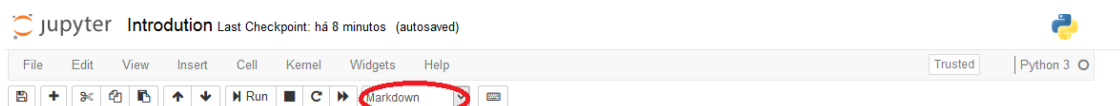


Figura 5 - Tipo de Célula

- i. Para remover uma célula, é só seleccionar a célula em questão, ir na aba “Edit” e clicar na opção “Delete Cells”.

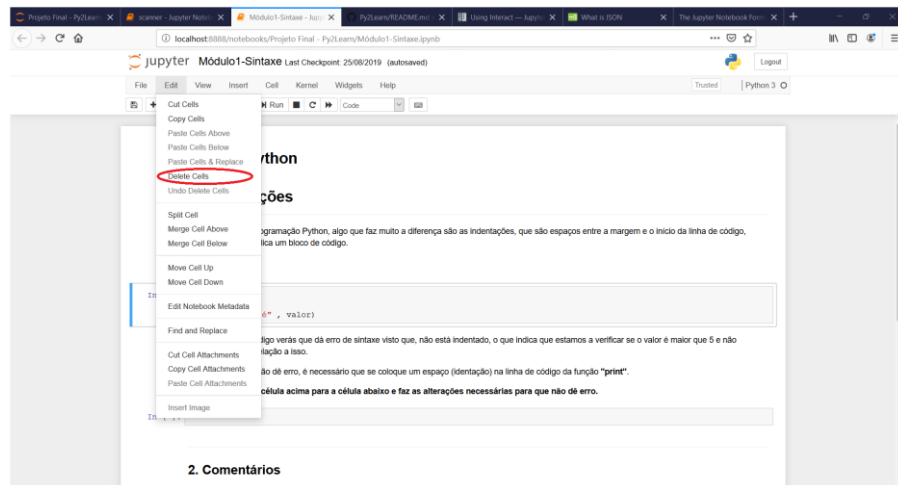


Figura 6 - Apagar Célula

- ii. Para guardar as alterações é só carregar no ícone do guardar.

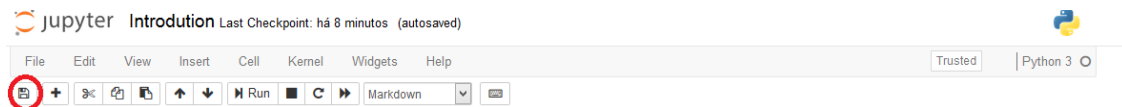


Figura 7 - Guardar Notebook

- iii. Caso queiras baixar para o teu computador o teu notebook alterado, na aba "File" podes escolher a opção "Download as" e de seguida "Notebook"

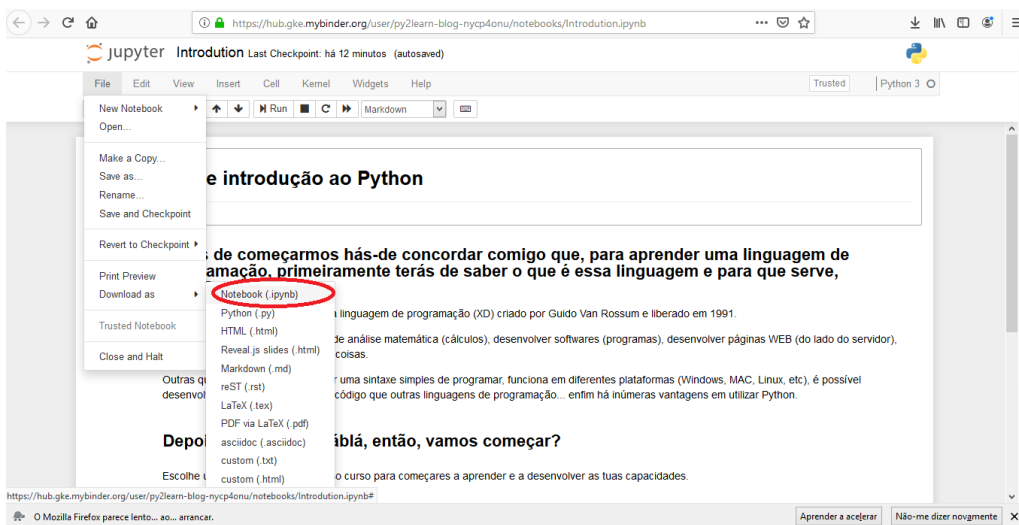


Figura 8 - Baixar Notebook

3.3 Abordagem

Visto que os notebooks ao serem compartilhados não possuem uma navegação muito facilitada, por se encontrarem dispostos por listas e se encontrarem num misto de tipos de conteúdos de forma que nenhum programador web gostaria de disponibilizar, foi necessário recorrer a ferramentas que permitissem melhorar um pouco mais a navegação do utilizador.

Inicialmente foram encontradas algumas opções que poderiam ser a solução para este grande problema, que era criar um *Jupyter Book*, que é uma extensão do *Jupyter* que permite disponibilizar os notebooks todos juntos num só livro, como se fossem páginas. O mesmo permitiria melhorar a navegação entre os notebooks, visto que possui uma barra lateral com o índice do livro e que ao clicar num elemento desse índice, seria direcionado para essa página.

No entanto essa solução não se tornou possível devido a um erro na criação das páginas de GitHub (algo indispensável), erro esse que não é pormenorizado a sua resolução nos passos da criação dos *Books*. Mas, no entanto, na imagem seguinte (Figura 19), é possível ter a noção de como ficaria tudo.

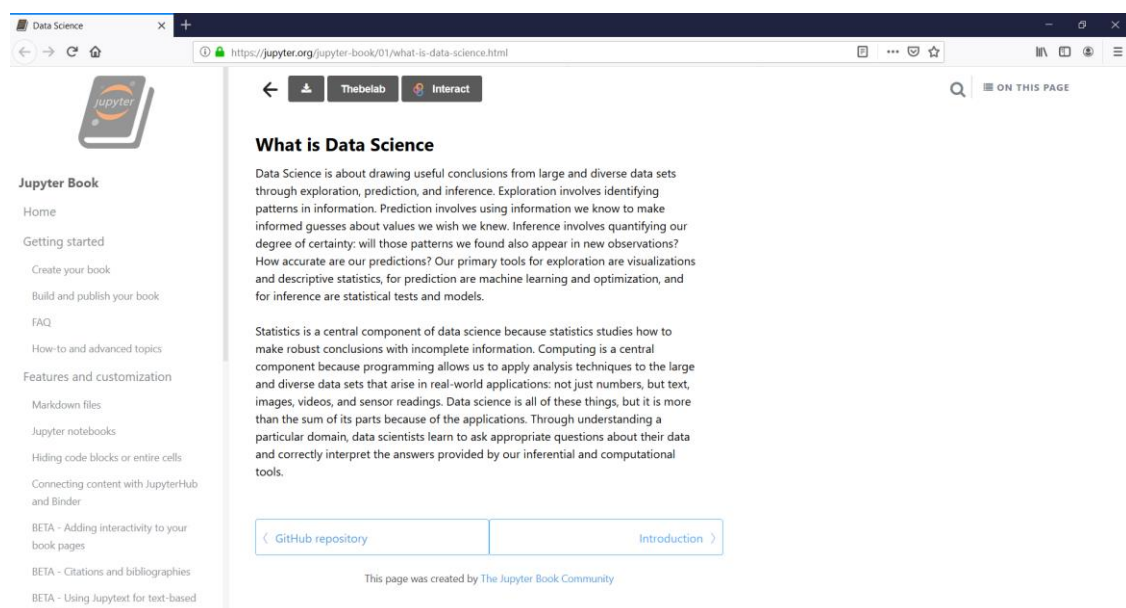


Figura 9 - Jupyter Book

De forma a permitir com que os utilizadores tivessem acesso ao curso, foi utilizado o Binder que permite através do link do repositório de notebooks do GitHub, criar um link que pode ser partilhável por um número ilimitado de utilizadores criando várias cópias do repositório. Desta forma ninguém poderia alterar o conteúdo principal do curso (a não ser da sua própria cópia) nem interferir com aquilo que outros vão fazendo.

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

Para conseguir ter o mínimo de sucesso na criação deste curso, é necessário dar alguns passos, como:

1. Entender como funciona exatamente o *Jupyter*;
2. Obter conteúdos teóricos para os módulos, de acordo com os temas que pretendo desenvolver;
3. Criar notebooks para cada módulo e introduzir os conteúdos;
4. Entender como posso partilhar os meus notebooks online, sem que o original seja alterado por outros;
5. Criar testes para os módulos;
6. Fazer testes finais;

4.1.1 Funcionamento do *Jupyter*

No capítulo anterior, é possível encontrar esse passo de uma forma detalhada, aonde se pode encontrar os tipos de células, entre outras funcionalidades dentro de um notebook da *Jupyter*.

No entanto para criar os notebooks, previamente tive a necessidade de instalar a plataforma [Anaconda](#), que é a plataforma que disponibiliza a aplicação do *Jupyter*.

Para instalar é só ir ao site da [Anaconda](#) e instalá-lo, de seguida após ser finalizada a instalação, devemos abrir o navegador da Anaconda e clicar no “Launch” do “Notebook”, como poderá ver na imagem abaixo.

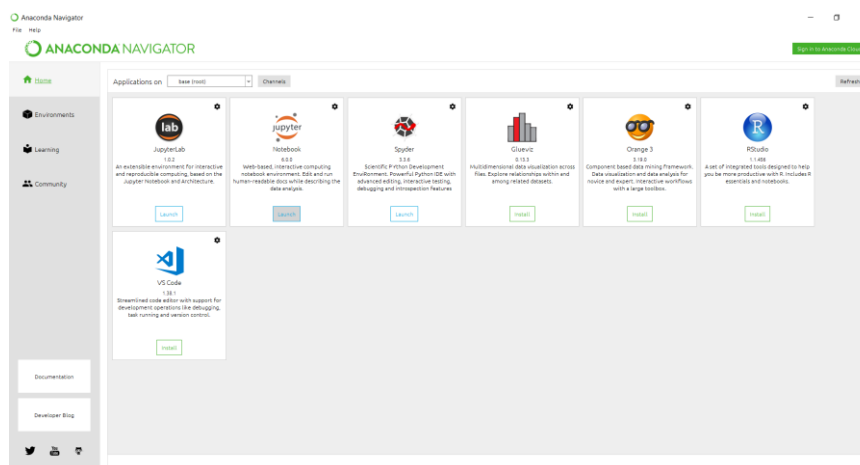


Figura 10 - Navegador do Anaconda

4.1.2 Obter Conteúdos e Criar os Notebooks

Os temas e os seus conteúdos foram pesquisados em diversos sites que especificavam 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.

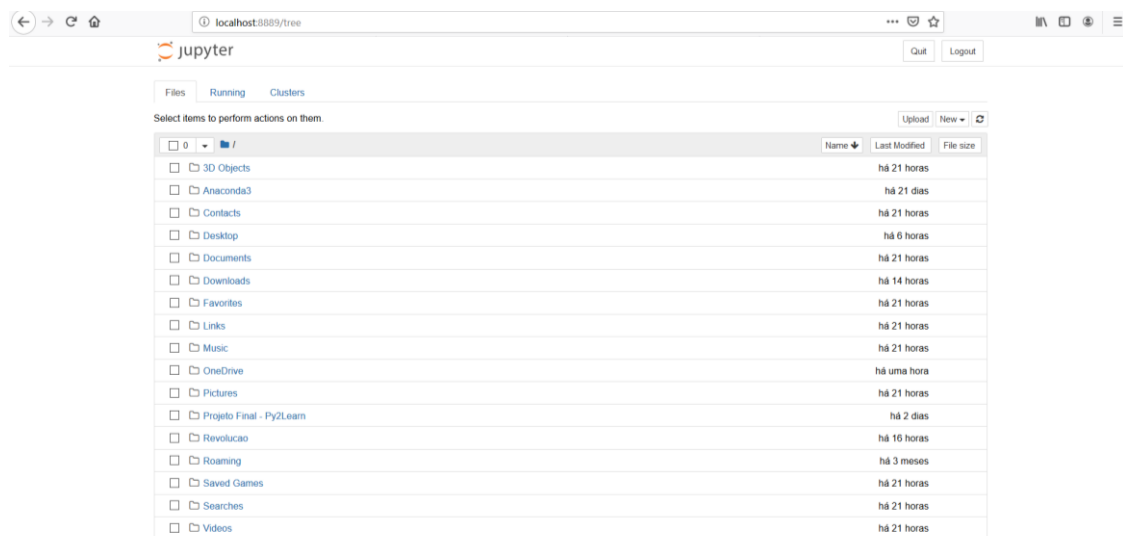


Figura 11 - Diretoria do Jupyter

Aquilo que temos de fazer é ir ao “New” e ou criar uma pasta para lá colocarmos os nossos notebooks, ou criamos diretamente o nosso notebook e começamos a introduzir lá os conteúdos que queremos que sejam apresentados nos nossos módulos. Respeitando os tipos de células de acordo com o conteúdo do capítulo 3.2.2.

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.

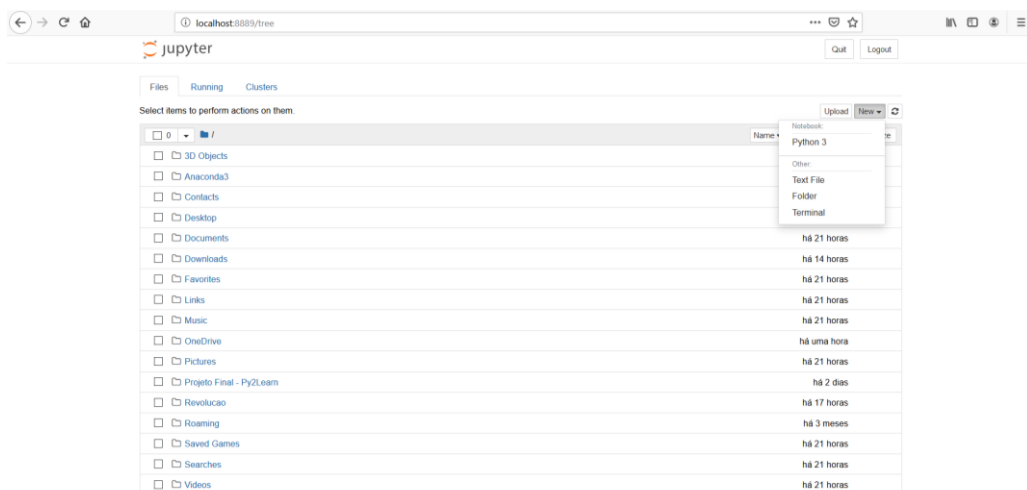


Figura 12 - Criar Notebook

4.1.3 Disponibilizar os Notebooks Online

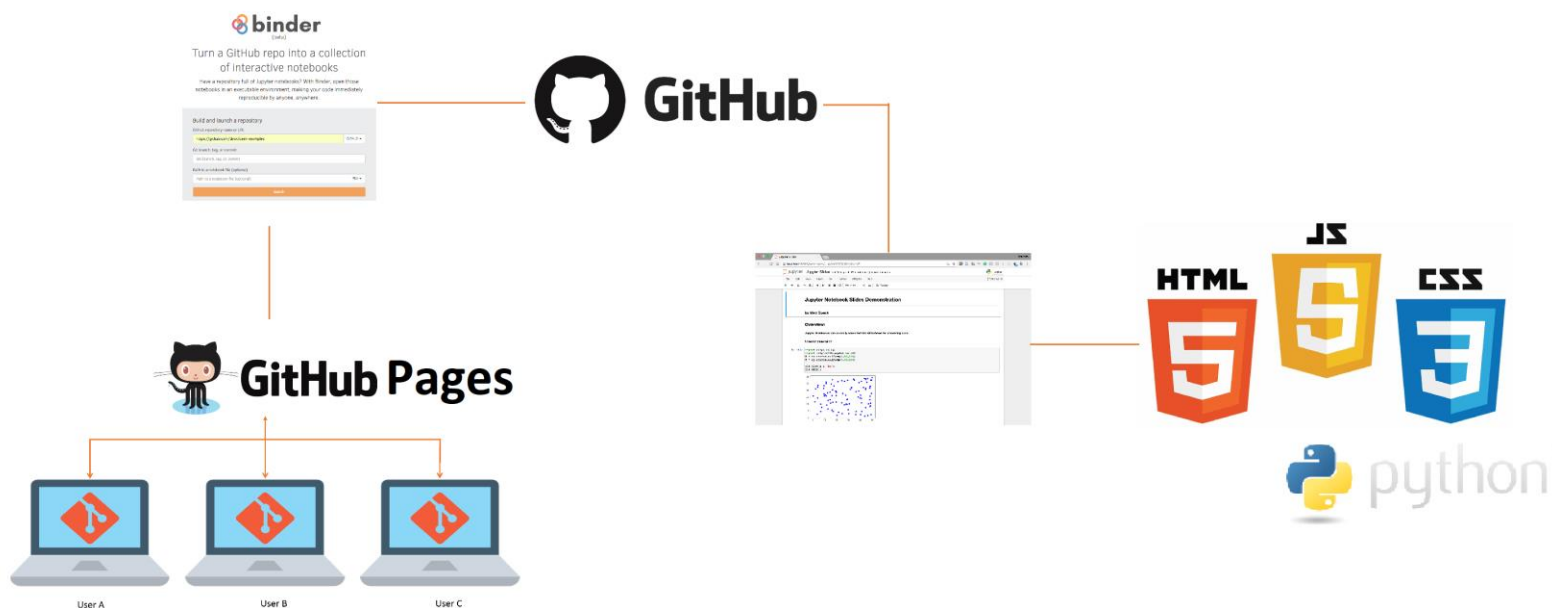


Figura 13 - Modelo de Funcionamento do Curso

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, nessa página ele terá acesso direto ao link dos notebooks (com todos os conteúdos do curso, incluindo os testes) hospedados no repositório da GitHub, link esse que foi gerado através do Binder.

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;

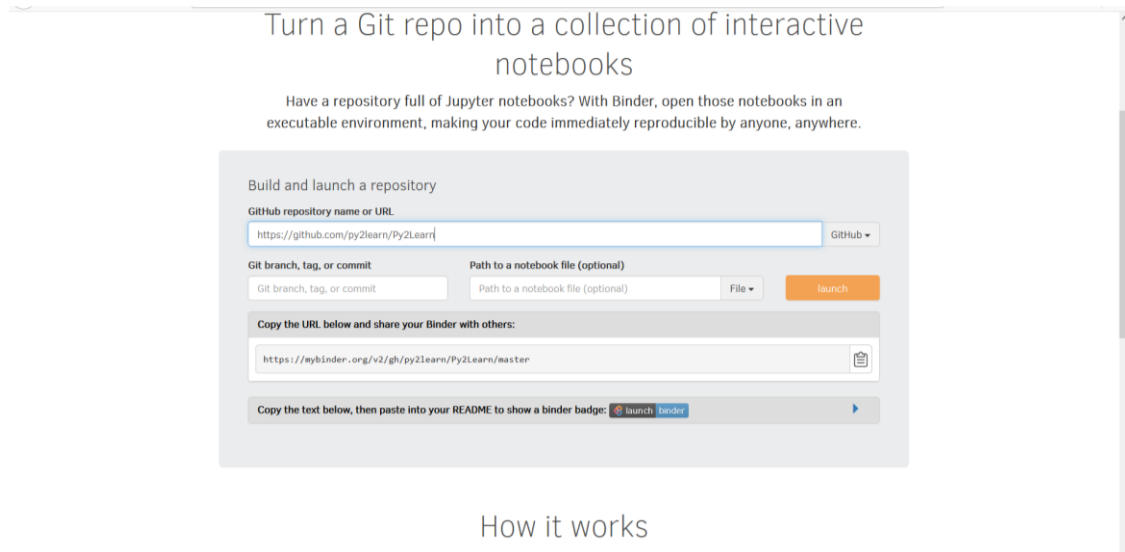


Figura 14 - Inserir link do GitHub no Binder

2. Clicar no botão “Launch”

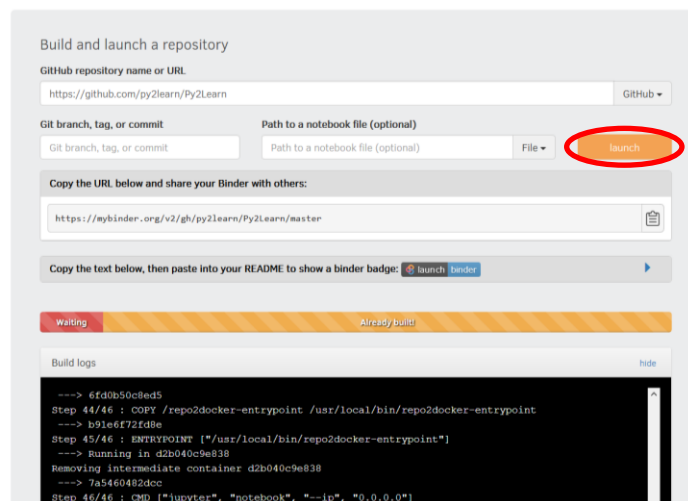


Figura 15 - Executar Repositório no Binder

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

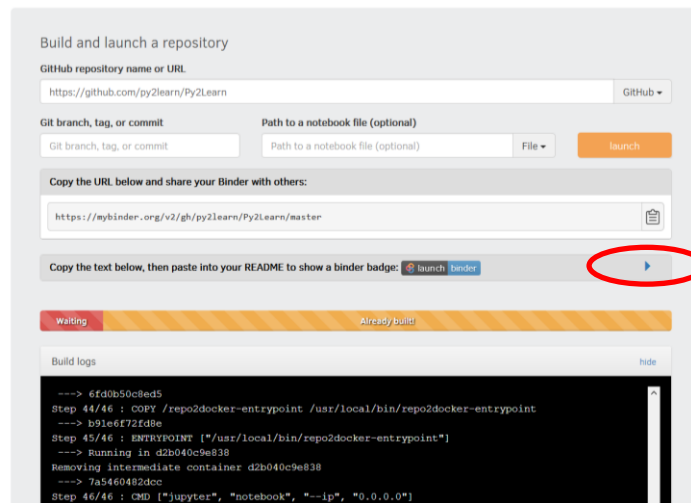


Figura 16 - Copiar Link do Binder

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

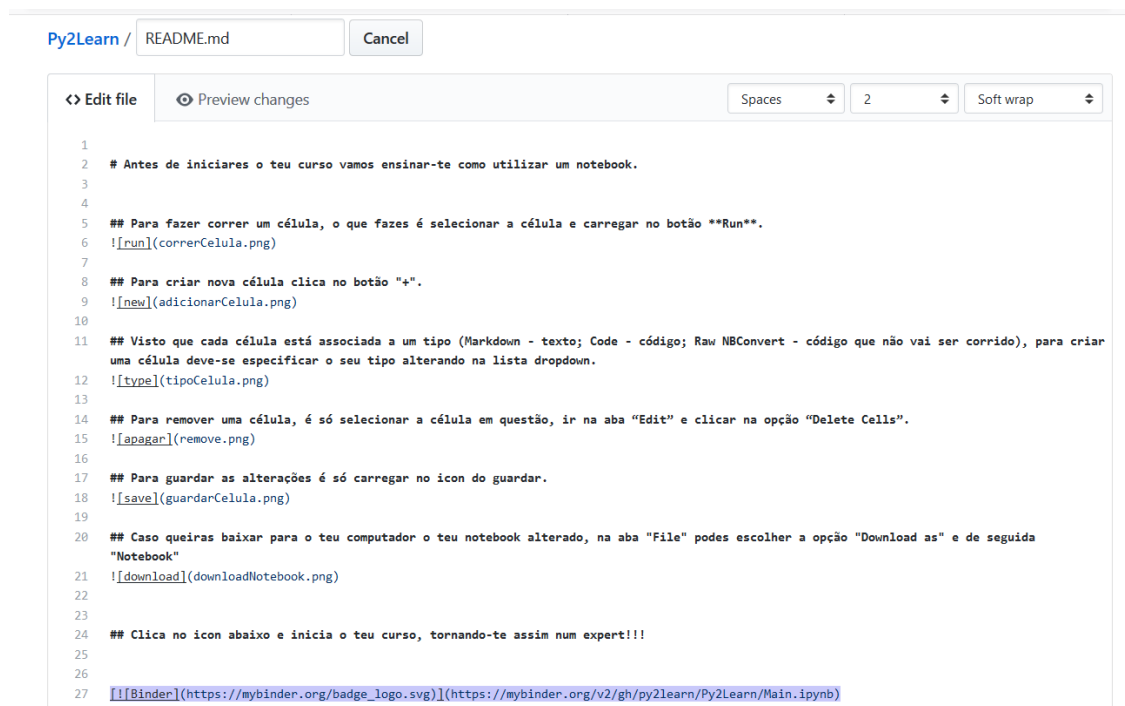


Figura 17 - Adicionar Link no README.md

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

4.1.4 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, ele se encontra completamente à vontade. 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:

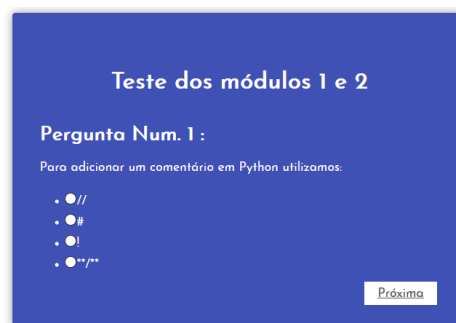
```
HTML( '''<html><javascript>'''código  
javascript'''</javascript></html>''' )
```

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 faz-se 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](#).

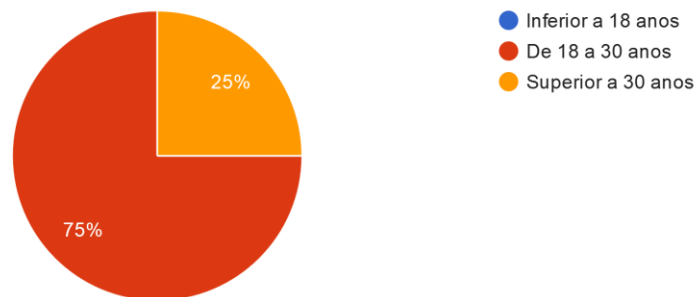
Figura 21 - Design do Teste

5 Validação e Testes

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

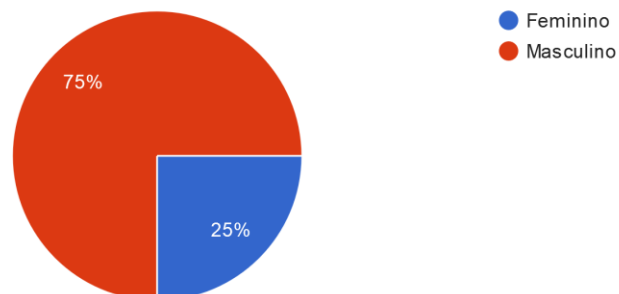
Idade

4 respostas



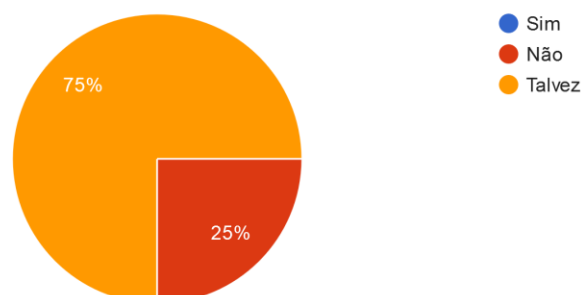
Sexo

4 respostas



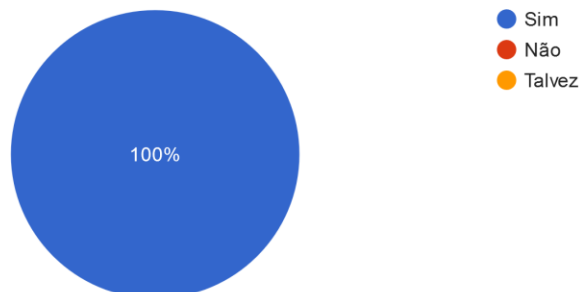
Foi fácil navegar entre os notebooks?

4 respostas



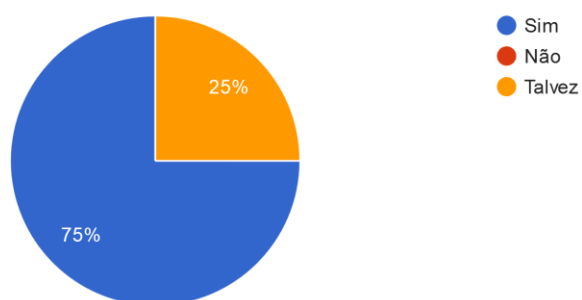
O curso é de fácil interacção?

4 respostas



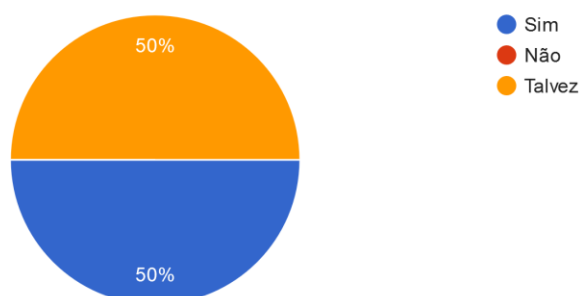
Foi fácil aprender os conteúdos?

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

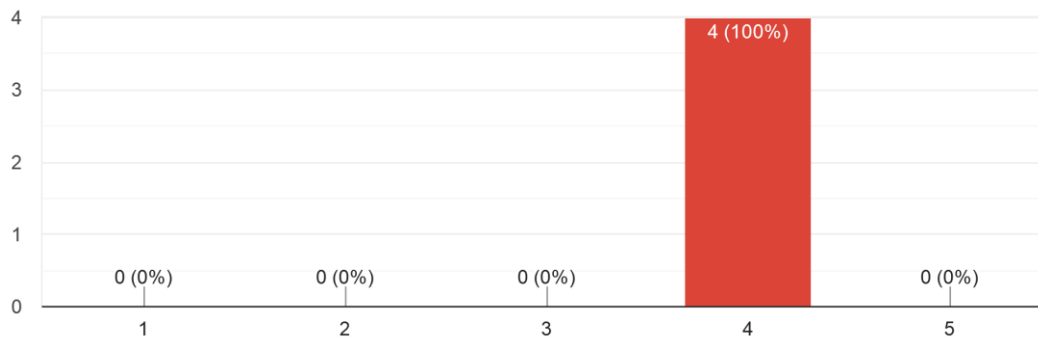
navegabilidade e proposta de sequência

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

Ter páginas de teste de conhecimento entre módulos

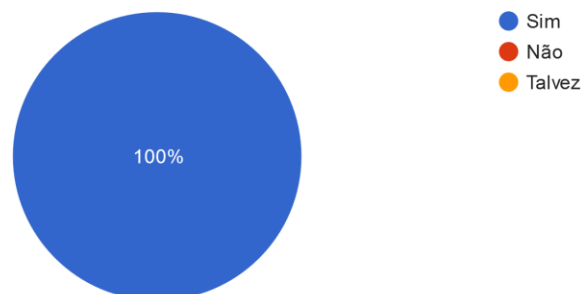
Acha este curso útil?

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 registrados 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.