



[VER PLANOS](#)

[PROGRAMAÇÃO](#)

[FRONT-END](#)

[DATA SCIENCE](#)

[INTELIGÊNCIA ARTIFICIAL](#)

[DEVOPS](#)

[UX & DESIGN](#)

[MOBILE](#)

[INOVAÇÃO & GESTÃO](#)

Artigos > **Data Science**





[VER PLANOS](#)

ao GitHub



Millena Gená Pereira

30/07/2021

[COMPARTILHE](#)



Esse artigo faz parte da
Formação Data Science

Google **colab** +  **GitHub**





VER PLANOS

deixados disponíveis para que recrutadores consigam ter fácil acesso e crucial para que conheçam um pouco mais sobre suas habilidades e expertises.

Pensando nisso, neste artigo iremos entender como cientistas de dados que utilizam o [Google Colaboratory](#) para o desenvolvimento de projetos de Data Science podem compartilhar suas análises e montar seu portfólio utilizando a plataforma de hospedagem de código, o GitHub.

Confira neste artigo:

- [Criando um repositório](#)
- [Editando o README](#)
- [Salvando o notebook do Google Colab no GitHub](#)
- [Criando uma pasta no repositório](#)
- [Salvando as alterações](#)

Criando um repositório

Após criar sua conta no GitHub, a primeira etapa para conseguir compartilhar seu notebook é criar um repositório. **O repositório vai ser o local no seu GitHub onde seu projeto ficará armazenado.** Normalmente cria-se um novo para cada projeto diferente.

Para criar um repositório, ao acessar sua conta, você pode clicar no ícone do seu perfil localizado no canto superior direito da tela e selecionar "Your Repositories":





VER PLANOS

Signed in as
millenagena

Set status

our pins

Your profile

Your repositories (highlighted with a red arrow)

Your codespaces

Your organizations

Your projects

Your stars

Your gists

Settings

Em seguida, você deve ser direcionado(a) para uma tela com os seus repositórios já existentes. Para criar um novo repositório, selecione o ícone verde escrito “New”, localizado na parte superior dessa aba:

Overview Repositories 3 Projects Packages

Find a repository... Type Language Sort

New (highlighted with a red box)

Feito isso, aparecerá uma tela onde você deverá nomear o repositório que deseja criar. Você pode nomear como achar mais interessante. O importante é que, **ao escrever o nome do repositório, não tenha nenhum espaço nem caracteres especiais**, uma vez que esse nome fará parte da URL para acessar seu repositório. Vou nomear o meu repositório de “Bootcamp_Data_Science_Alura”, porque nele eu irei adicionar os notebooks que foram desenvolvidos durante o nosso Bootcamp de Data Science:





VER PLANOS

Import a repository.

Owner *



millenagena ▾

Repository name *

Bootcamp_Data_Science_Alura



Great repository names are short and memorable. Need inspiration? How about [reimagined-garbanzo](#)?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file

This is where you can write a long description for your project. [Learn more](#).

Add .gitignore

Choose which files not to track from a list of templates. [Learn more](#).

Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

This will set main as the default branch. Change the default name in your [settings](#).

Create repository

Após nomeado, é importante não esquecer de marcar a checkbox descrita como “Add a README file”. Com essa opção marcada, ao criar seu repositório, automaticamente será gerado um arquivo do tipo MD (Markdown). Esse arquivo README.md será utilizado posteriormente para colocar uma descrição a respeito do projeto que se encontra naquele repositório.

Em seguida, selecione a opção “Create repository” para fechar essa janela e ir para a página inicial do seu repositório:





[VER PLANOS](#)

A screenshot of a GitHub repository page. At the top, there's a header with a user icon and the text 'VER PLANOS'. Below the header, a list of files shows 'README.md' with a red arrow pointing to it. The main content area displays the file's content: 'Bootcamp_Data_Science_Alura'. There's also a small edit icon next to the file name.



Editando o README

Ao acessar a página inicial do seu repositório, perceba que o arquivo README foi criado. Para editá-lo, basta clicar no ícone do README.md e selecionar o símbolo de lápis que aparece do lado direito:

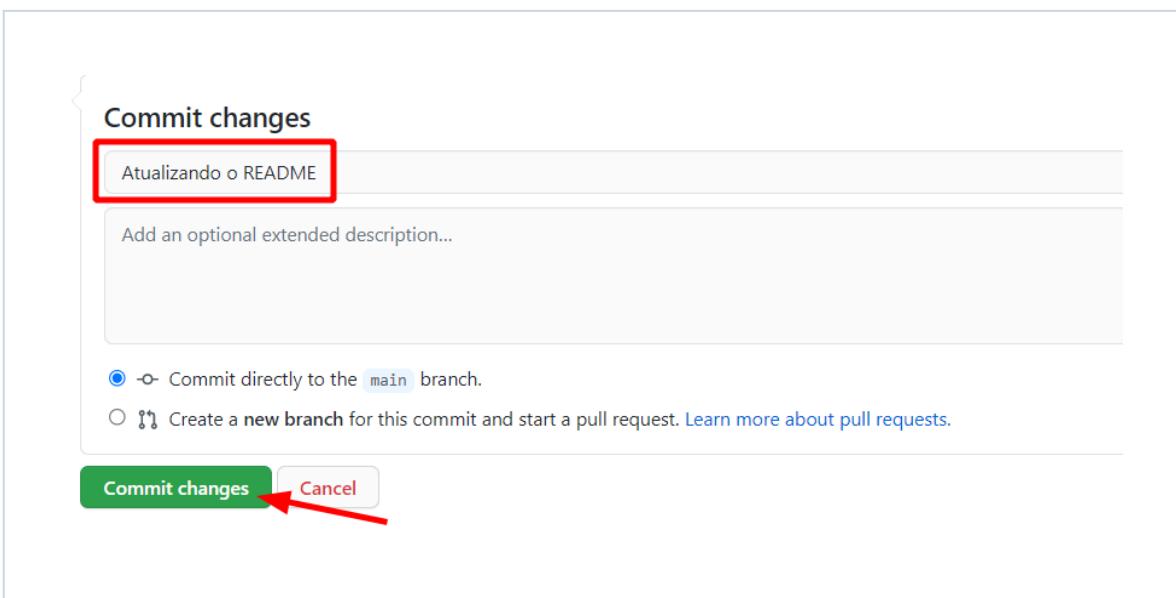
A screenshot of the GitHub README file editor. The top bar shows the repository owner 'millenagena' and the commit history. Below the editor area, there's a preview of the file content: 'Bootcamp_Data_Science_Alura'. On the right side of the editor, there are several icons: 'Edit this file' (with a pencil icon), 'Raw', 'Blame', 'Copy', and 'Delete'.





VER PLANOS

O README é o nosso primeiro contato ao acessar seu repositório. Por isso, é importante que ele seja detalhado e contenha um resumo sobre as partes do seu projeto. Depois de escrever as informações do README, ao rolar até o final desta página, você vai encontrar um botão denominado “Commit changes”. Um commit é realizado quando é feita alguma alteração no nosso projeto e é interessante que ele descreva de forma sucinta a alteração que foi realizada, por exemplo:



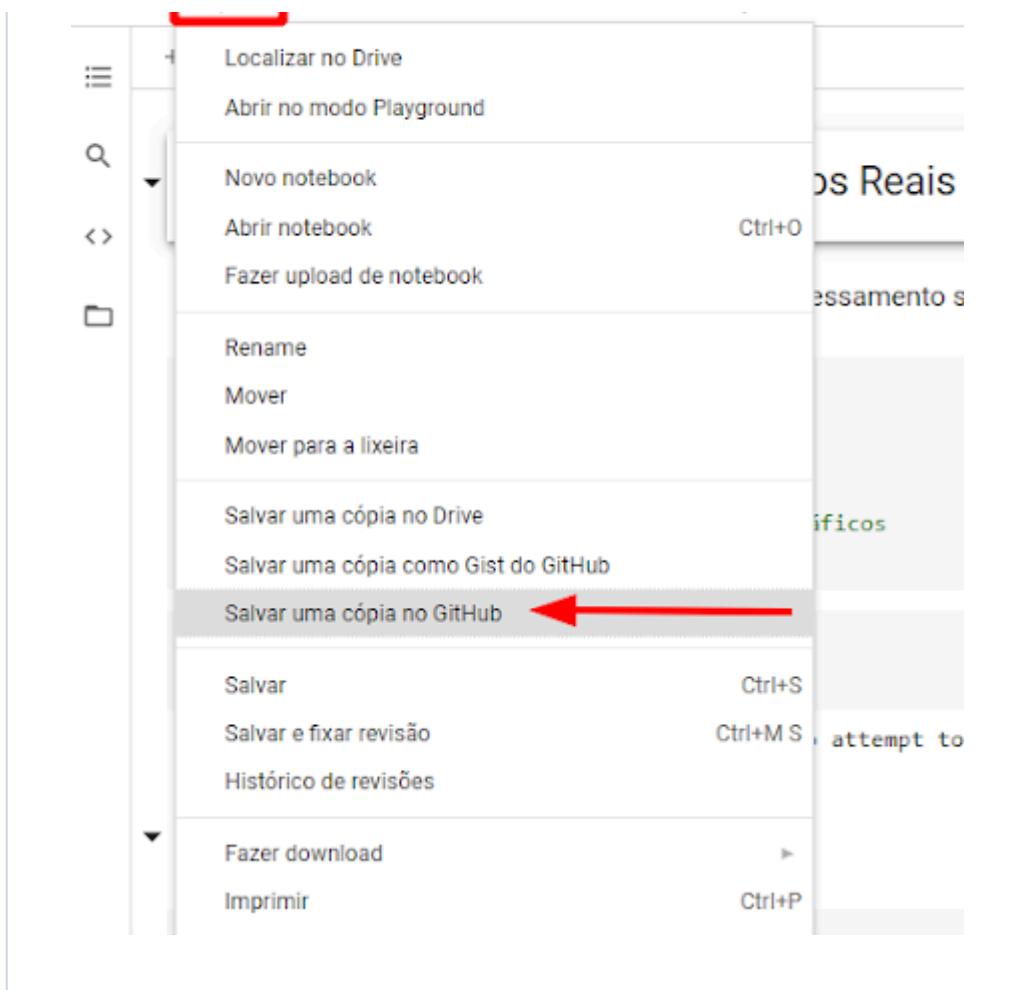
Salvando o notebook do Google Colab no GitHub

Agora, vamos salvar o notebook do Google Colab, que contém nosso projeto, no repositório que criamos. Para fazer isso é necessário acessar o notebook que deseja adicionar e executar todas as suas células para salvar qualquer modificação que tenha sido realizada. Após, na barra de ferramentas do Colab, você pode acessar **Arquivos > Salvar uma cópia no GitHub**:

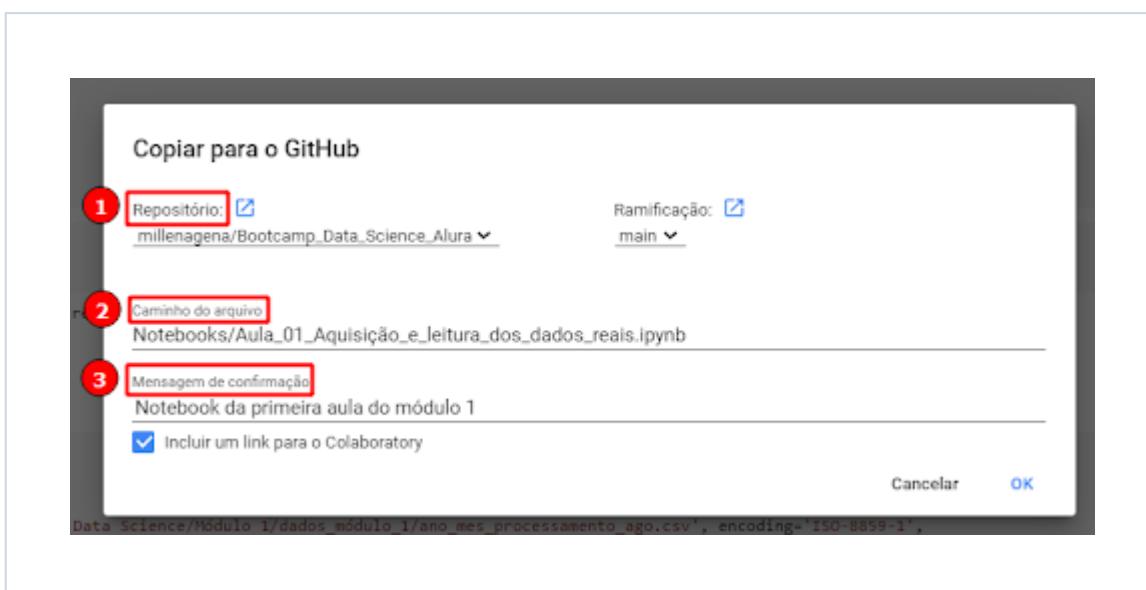




VER PLANOS



Caso seja a primeira vez que você realiza a conexão do Google Colab ao GitHub, o Colab deverá solicitar o acesso ao GitHub por meio do seu login e senha para que ele possa permitir que seus notebooks sejam salvos. Depois dessa etapa, deve aparecer uma tela denominada **Copiar para o GitHub**:





VER PLANOS

Onde deseja salvar seu notebook;

2. Caminho do arquivo: esse outro campo é designado para você selecionar em qual parte do seu repositório deseja adicionar esse arquivo. Caso você escolha adicionar apenas o arquivo, basta deixar apenas o nome dele. Mas, se desejar que ele fique dentro de alguma pasta específica do seu GitHub, você deve adicionar o nome dessa pasta antes do nome do arquivo e colocar uma “/”, seguindo a estrutura: **nome_da_pasta/nome_do_arquivo**;
3. Mensagem de confirmação: assim como o commit, mostrado anteriormente, a mensagem de confirmação é um campo no qual deve ser descrita a alteração que foi realizada no projeto.

Com esses campos preenchidos, ao clicar em “OK”, o notebook será salvo no repositório selecionado e dentro da respectiva pasta que foi criada por meio do “Caminho do arquivo”.

O GitHub normalmente disponibiliza uma prévia do seu notebook no próprio site mesmo. No entanto, ao acessar o repositório logo após ter salvo seu notebook, pode acontecer do carregamento dessa prévia não estar disponível ainda e aparecer um aviso indicando um erro. Mas, após algum tempo, ao sair e acessar novamente o site, logo essa prévia deve ser apresentada:

The screenshot shows a GitHub repository page for a notebook named "Notebook da primeira aula do módulo 1". The notebook has 1218 lines (1218 sloc) and 232 KB. It was last committed 6 minutes ago. The page includes a preview of the notebook content, which starts with importing libraries like seaborn and pandas. A "Open in Colab" button is visible at the top of the preview area.

Criando uma pasta no repositório





[VER PLANOS](#)

próprio GitHub, é necessário voltar para a página original do repositório e clicar em **Add file > Create new file**:

The screenshot shows a GitHub repository interface. At the top, there are buttons for 'main' (branch), '1 branch', '0 tags', 'Go to file', 'Add file', and 'Code'. Below these are three items listed: 'millenagena Notebook da primeira aula do módulo 1' (a file), 'Notebooks' (a folder), and 'README.md' (a file). On the right side of the list, there are 'Upload files' and '3 commits' buttons. A red arrow points to the 'Create new file' button, which is highlighted with a blue background.

Na próxima página, para criarmos a pasta “Dados” devemos digitar “Dados/”, junto com essa barra, para que o GitHub entenda que queremos criar uma pasta:

The screenshot shows a GitHub commit message input field. The path 'Bootcamp_Data_Science_Alura / Dados /' is entered, followed by a placeholder 'Name your file...' in a blue-bordered input field. To the right, it says 'in main'.

Agora, para que essa pasta seja devidamente criada é necessário criar um arquivo simbólico, para que o GitHub entenda que você está criando uma pasta na qual poderão ser adicionados arquivos. Como ele será excluído posteriormente, você pode nomeá-lo como preferir:

The screenshot shows a GitHub commit message input field. The path 'Bootcamp_Data_Science_Alura / Dados /' is entered, followed by the file name 'arquivo_simbolico.txt' in a blue-bordered input field. To the right, it says 'in main'.

Agora que a pasta já foi criada, para salvarmos os dados do nosso projeto nela, podemos acessar essa pasta e clicar em **Add files > Upload files**:

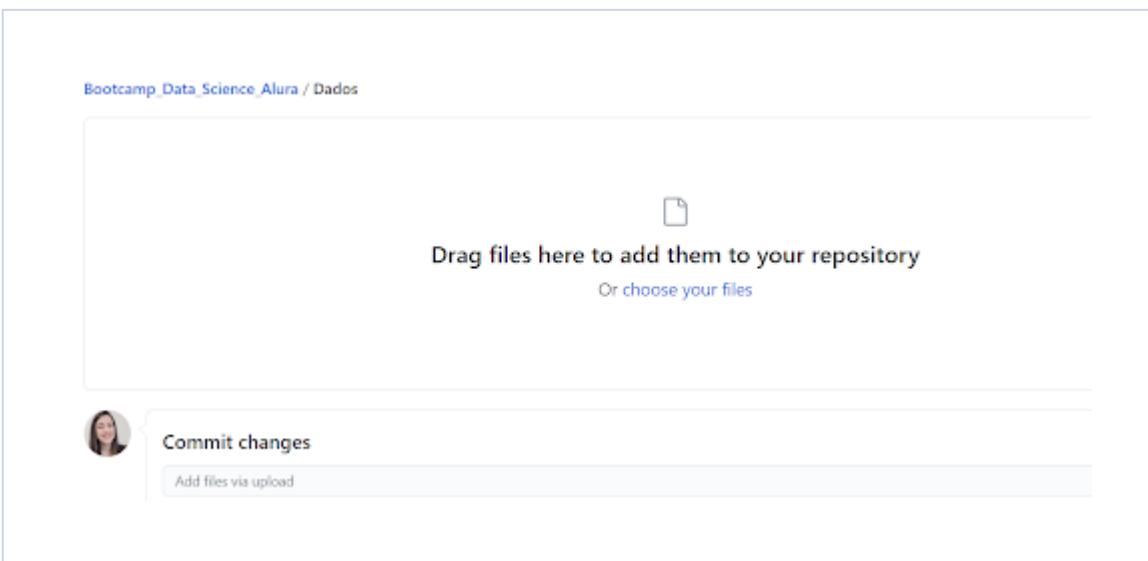




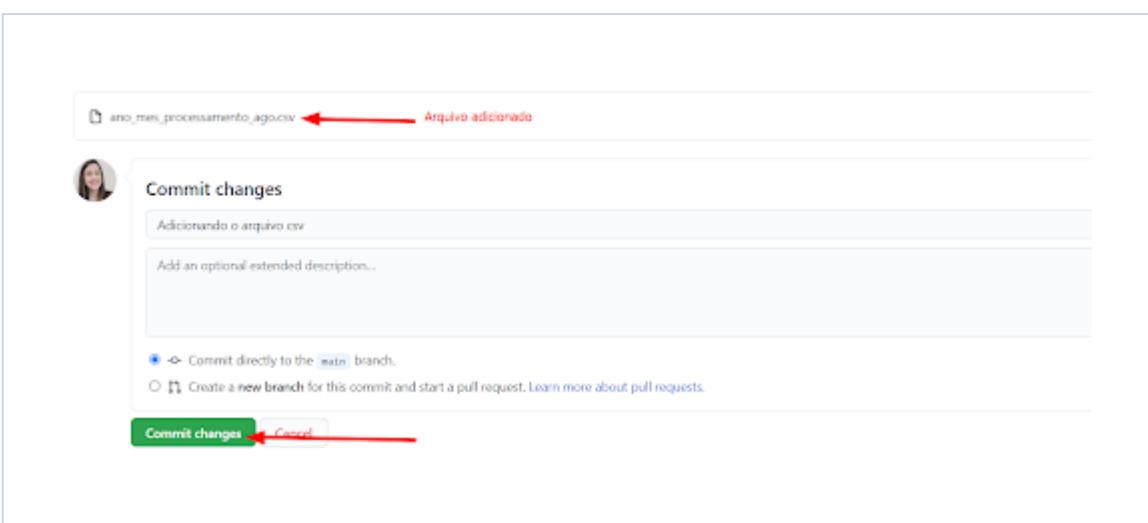
VER PLANOS



Em seguida, será apresentada uma página onde poderá arrastar o arquivo que deseja fazer upload ou selecionar a opção “choose your files”, para que selecione esse arquivo diretamente das pastas da sua máquina.



Com o arquivo selecionado, você pode escrever seu respectivo commit e clicar em “Commit changes” para salvar as alterações.



O "arquivo_simbolico.txt" anteriormente criado, agora pode ser excluído. Ele existia apenas para que a pasta “Dados” não ficasse vazia, uma vez que não tem como criarmos uma pasta vazia no GitHub. Sendo assim, **você só pode excluir o arquivo simbólico a partir do momento que existir outro arquivo na pasta**.





VER PLANOS

Salvando as alterações

Agora, nosso repositório está pronto e o projeto está salvo. Mas, e quando fizermos uma alteração no nosso notebook, como podemos salvar essa alteração também no nosso repositório?

É simples! Basta seguir aquele mesmo processo que realizamos anteriormente quando fomos salvar nosso notebook pela primeira vez no repositório. Você pode acessar **Arquivos > Salvar uma cópia no GitHub**. Na parte de “Caminho do arquivo” é importante colocar o mesmo caminho que colocamos anteriormente, com o nome da pasta e o mesmo nome do arquivo. No espaço onde está escrito “Mensagem de confirmação”, você pode escrever a alteração que foi realizada no notebook e clicar em “OK”.

Veja o exemplo:

Copiar para o GitHub

Repositório: [millenagena/Bootcamp_Data_Science_Alura](#) Ramificação: [main](#)

Caminho do arquivo
Notebooks/Aula_01_Aquisição_e_leitura_dos_dados_reais.ipynb Mesmo caminho anterior

Mensagem de confirmação
Importando a biblioteca Numpy

Incluir um link para o Colaboratory

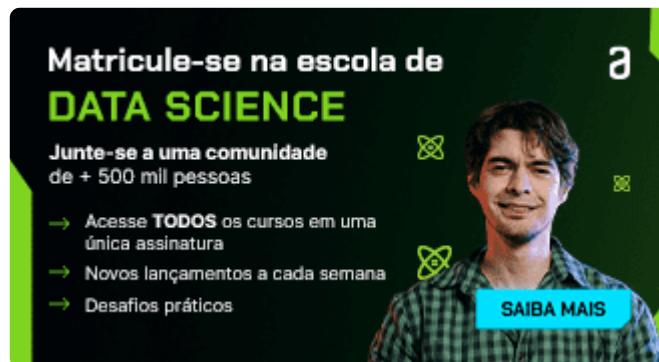
Cancelar OK

E pronto! Alteração salva e commit feito.



[VER PLANOS](#)

[seu código e Git e GitHub: Estratégias de ramificação, Conflitos e Pull Requests.](#)



Millena Gená Pereira

Millena Gená é estudante de Ciência da Computação. Atualmente, é Instrutora de Dados aqui na Alura atuando principalmente na área de Engenharia de dados. Ela está sempre procurando aprender algo novo sobre tecnologia e é apaixonada por novas aventuras.

Programar e ajudar as pessoas são seus hobbies favoritos! ^^

[Artigo Anterior](#)

[Próximo Artigo](#)

[Data Lake vs Data Warehouse](#)

[Big Data: Entenda mais sobre esse conceito](#)

Leia também:

- [Google Colab: o que é, tutorial de como usar e criar códigos](#)
- [Engenharia de Dados: o que é, o que faz e um Guia completo](#)
- [Power BI: o que é, para que serve, quem utiliza e um Guia para iniciar](#)





[VER PLANOS](#)

Veja outros artigos sobre
[Data Science](#)

Quer mergulhar em tecnologia e aprendizagem?

Receba conteúdos, dicas, notícias, inovações e tendências sobre o mercado tech diretamente na sua caixa de entrada.

Email*

[ENVIAR](#)

Nossas redes e apps





VER PLANOS

Carreiras Alura

Como Funciona

Para Empresas

Todos os cursos

Para Sua Escola

Depoimentos

Política de Privacidade

Instrutores(as)

Compromisso de Integridade

Dev em <T>

Termos de Uso

Luri, a inteligência artificial da Alura

Documentos Institucionais

IA Conference 2024

Status

Conteúdos

Alura Cases

Fale Conosco

Email e telefone

Imersões

Perguntas frequentes

Artigos

Podcasts

Artigos de educação

corporativa

Novidades e Lançamentos

Email*

ENVIAR





VER PLANOS

Cursos de Programação

Lógica | Python | PHP | Java | .NET | Node JS | C | Computação | Jogos | IoT

Cursos de Front-end

HTML, CSS | React | Angular | JavaScript | jQuery

Cursos de Data Science

Ciência de dados | BI | SQL e Banco de Dados | Excel | Machine Learning | NoSQL | Estatística

Cursos de Inteligência Artificial

IA para Programação | IA para Dados

Cursos de DevOps

AWS | Azure | Docker | Segurança | IaC | Linux

Cursos de UX & Design

Usabilidade e UX | Vídeo e Motion | 3D

Cursos de Mobile

React Native | Flutter | iOS e Swift | Android, Kotlin | Jogos

Cursos de Inovação & Gestão

Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas

CURSOS UNIVERSITÁRIOS FIAP

Graduação | Pós-graduação | MBA

