

Escopo do Projeto

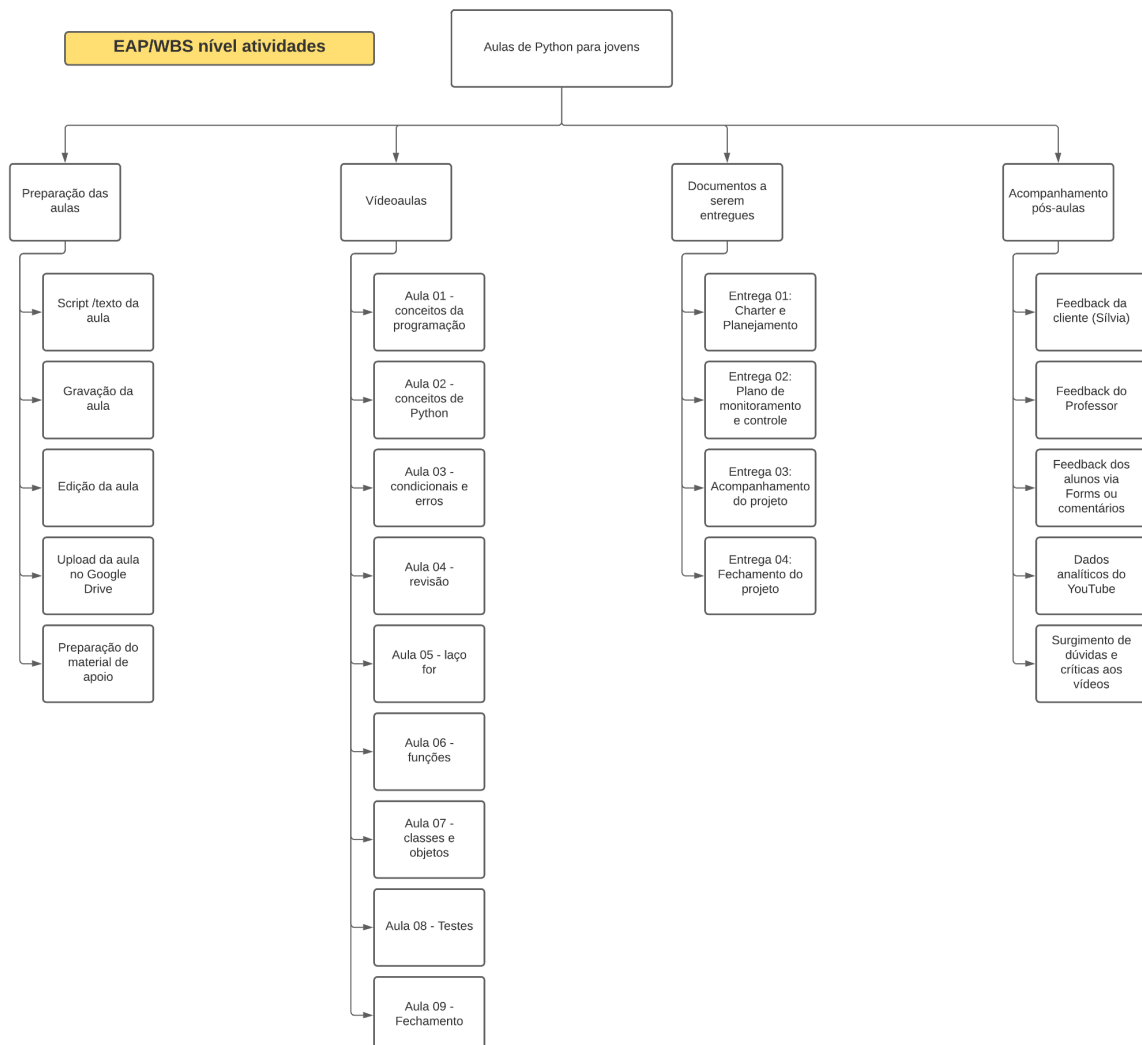
TT060B

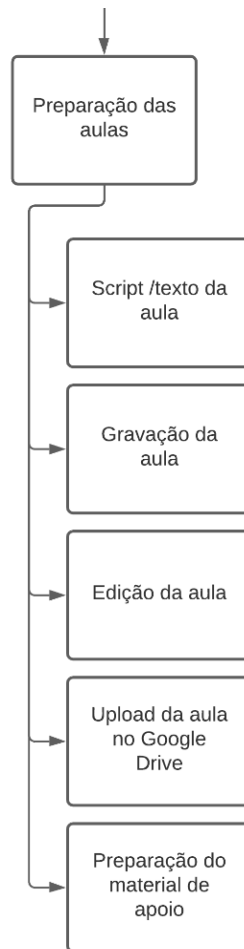
Integrantes:

Arthur Felipe (Gerente) - 231661
Eric Camille Camargo dos Santos - 215419
Matheus Eduardo Da Silva - 230719
Verônica Cintra de Oliveira - 244963

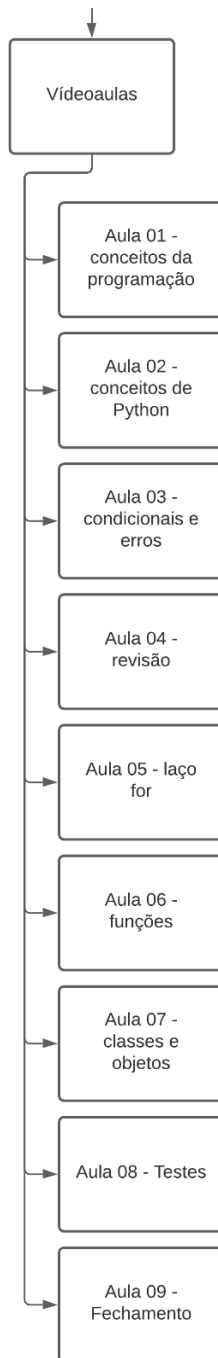
Escopo do projeto

Para detalhar o escopo do projeto, foram feitos dois modelos de EAP/WBS: um detalhando as atividades que devem ser executadas durante o projeto (o qual pode ser observado logo abaixo) e outro que detalha pelo que cada um dos integrantes será responsável.

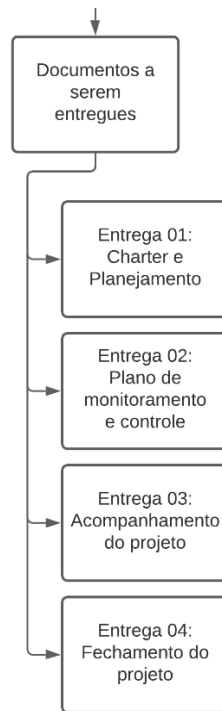




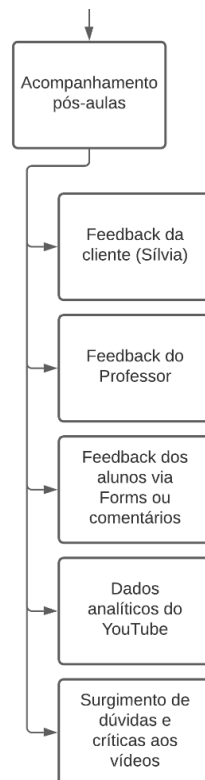
Em “Preparação das aulas”, temos as atividades que devem ser constantemente repetidas antes de cada aula gravada.



Em “Videoaulas”, temos os deliverables do LIAG/Sílvia



Em “Documentos a serem entregues”, temos os deliverables do Professor



E, em “Acompanhamento pós-aula”, temos descrito todo o acompanhamento que o grupo fará dos vídeos entregues. Salientamos que, apenas o feedback do Professor, da Sílvia e o surgimento de dúvidas serão considerados como **escopo** do projeto. Feedback via Forms, comentários e ‘Analytics’ do YouTube serão um trabalho extra feito pelo grupo.

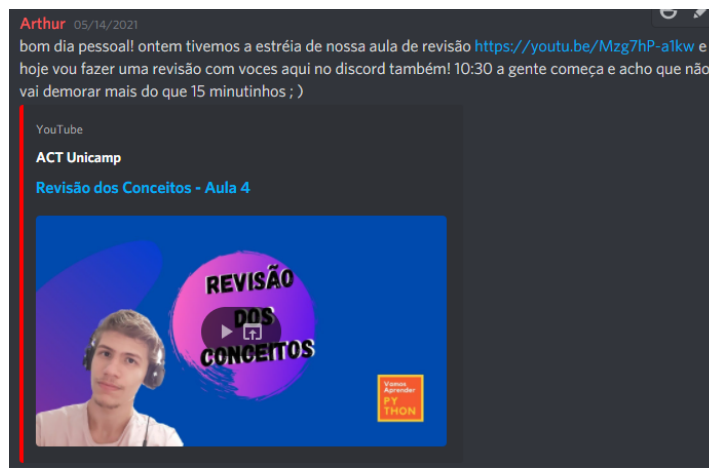
Detalhamento do Escopo das Aulas

As videoaulas serão separadas em:

- **1º vídeo - Introdução:** Neste vídeo, falaremos sobre os conceitos base da programação ('o que é programar?'). O vídeo será subdividido em 3 partes de 28 minutos cada. Na primeira parte, vamos falar sobre o que é possível fazer com a programação (onde que a programação está? Como afeta nossas vidas? O que e como a programação mudou nossas vidas?); a ideia é que, ao aproximar o aluno dos motivos pela programação ser hoje o que ela é, eles ganharão mais ânimo em aprender mais sobre este universo. Na segunda parte, vamos tratar sobre as diferenças entre conversar com uma pessoa e 'conversar' com um computador, explicando como ele não entende frases comuns e precisamos de algumas ferramentas que tornam essa 'conversa' algo possível. E, por fim, na terceira parte, vamos fazer um pequeno exercício sobre lógica de programação, onde o apresentador vai incentivar os alunos a pensarem em um conjunto de direções que o apresentador precisa seguir para conseguir encontrar um aplicativo em seu próprio celular (por exemplo, procure o botão de liga (mas O QUE É um botão de desliga?)->desbloqueie (como que eu desbloqueio?)->procure o aplicativo (até onde eu vou passar as páginas até encontrar o aplicativo?)->clique no aplicativo (qual é o nome do aplicativo? preciso deixar isso armazenado em algum lugar)), direcionando os jovens a pensar com uma lógica de programação.
 - Desafio da aula: Procurar qual foi a primeira linguagem de programação e qual é a linguagem de programação mais utilizada hoje em dia.
- **2º vídeo - Entendendo a lógica dos computadores:** Após fazer uma breve introdução falando superficialmente sobre FORTRAN e Python (relacionados à pesquisa do vídeo anterior), neste vídeo, começaremos lembrando o joguinho mental que realizamos no último vídeo e os principais conceitos que tratamos anteriormente (para que, mesmo se alguém perdeu o vídeo anterior, consiga entender minimamente do que estamos falando. Não queremos deixar ninguém pra trás). Após, vamos falar sobre linguagens de programação, o que são linguagens de programação e como funcionam (de onde vieram, para que servem, quais existem e como usamos [ou seja, suas aplicações]). Aproveitaremos este momento para corrigir a atividade da aula anterior, falando superficialmente sobre FORTRAN (primeira linguagem de programação disponível) e Python (a mais comum atualmente), até que, por fim, vamos falar mais à fundo de Python, dando exemplos da linguagem (principalmente através de "prints", inputs e variáveis, conceitos que utilizaremos para mostrar em linguagem de programação o jogo mental que fizemos na aula anterior, de encontrar um aplicativo).
 - Desafio da aula: Entrar no link do Google Colab disponibilizado para apenas executar cada um dos comandos ali disponíveis.
- **3º vídeo - Caçando problemas:** Neste vídeo, vamos focar em encontrar problemas. Vamos começar com um simples "jogo dos 7 erros", onde teremos diversas imagens para os alunos encontrarem os erros e, ao passar das imagens, a dificuldade em encontrar os erros vai aumentando. A ideia é acostumar os alunos

com a busca por inconsistências e coisas fora do padrão. Em seguida, mostraremos alguns exemplos de códigos como os feitos no vídeo anterior, um correto, e o outro, com erros de digitação, indentação e/ou semântica, buscando ensinar os alunos da importância da busca por correteza durante a escrita de código e da importância da refatoração. Por fim, vamos fazer um programa que lê um “input” do usuário e retorna um dado processado, contendo, durante o desenvolvimento da atividade, alguns erros propositais de indentação e digitação, mostrando aos alunos como que erros acontecem, como os corrigimos, ao mesmo tempo que mostramos recursos mais aprofundados da linguagem.

- Desafio da aula: Entrar no link do Google Colab disponibilizado para analisar os códigos, encontrar os erros e corrigi-los. Os alunos devem enviar as respostas via Discord ou via E-mail.
- **4º vídeo - Condicionais, refatoração e erros:** Vamos começar a aula corrigindo os exercícios deixados como tarefa na aula anterior. Após, vamos falar sobre os erros e warnings. Para encerrarmos, vamos começar a discutir laços de repetição, dando exemplos práticos de repetições que fazemos diariamente e iremos mostrar na prática como isso pode ser utilizado no universo da programação.
 - Desafio da aula: Entrar no link do Google Colab disponibilizado para analisar os códigos e “otimizá-los”, utilizando laços de repetição. Serão exercícios bem simples, onde implementaremos apenas um laço “for”. Os alunos devem enviar as respostas via Discord ou via E-mail.
- **Revisão Discord:** Na sexta-feira seguinte à 4ª aula, fizemos uma reunião com os integrantes do discord para corrigir ao vivo os exercícios extra da aula.



- **5º vídeo - Revisão e repetição:** Este vídeo será dividido em duas partes: na primeira, vamos revisar os conceitos base vistos até então (como print, variáveis, listas, erros, warnings, etc.) utilizando diversos exemplos simples de código em Python para demonstrar seu caso de uso mais comum. Na segunda parte vamos continuar a discussão a respeito de laços de repetição, explicando o que é um laço de repetição, qual a ideia por trás de um laço de repetição e dar exemplos mais 'palpáveis' (ex.: ‘Execute determinada tarefa até uma meta ser atingida. Assim que a meta for atingida, pare.’). Seguiremos com exemplos em Python, para exemplificar seu uso detalhado

- Desafio da aula: Entrar no link do Google Colab disponibilizado para analisar os códigos que contêm laços de repetição e corrigi-los. Os alunos devem enviar as respostas via Discord ou via E-mail.
- **6º vídeo - Funções:** Este vídeo será dividido em duas partes: na primeira, vamos abordar quais são os fundamentos de uma função: o que são, como funcionam, para que servem e como declaramos uma função. Após, vamos para o Google Colab para analisarmos alguns códigos que incluem funções, implementando uma função que será responsável por iterar sobre uma lista em Python, com o objetivo de poupar trabalho do desenvolvedor em implementar diversos laços 'for' e reduzindo código. Conforme mostramos o exemplo, incluiremos também exemplos de erros recorrentes, casos de uso e implementação.
 - Desafio de aula: Entrar no link do Google Colab disponibilizado para fazer os exercícios de teste disponibilizados. Os alunos devem enviar as respostas via Discord ou via E-mail.
- **7º vídeo - Classes e objetos:** Neste vídeo, vamos começar abordando os conceitos mais importantes de Classes e objetos, como o que é um objeto, o que significa "um objeto ser a instância de uma classe", para que e onde estão presentes, etc. A ideia desta aula é tornar os alunos familiarizados com os conceitos de programação orientada a objetos, entendendo principalmente os conceitos de herança e abstração . Após, vamos entrar no Colab para mostrar aos alunos um simples exemplo em Python para exemplificar os conceitos vistos.
 - Esta aula não terá nenhum desafio, apenas a apostila desenvolvida pela equipe para auxiliar os alunos.
- **8º vídeo - Testes:** Este será um vídeo mais curto, especialmente por que os testes são conceitos que ficam mais consolidados uma vez que os alunos já têm alguma prática com o desenvolvimento de software, mas, estaremos focando no principal, como o que é um teste, quais são os tipos de teste (teste de mesa, funcional e de usabilidade), como fazer um teste e por que devemos testar um programa.
 - Esta aula não terá nenhum desafio, apenas a apostila desenvolvida pela equipe para auxiliar os alunos.
- **9º vídeo - Análise de dados:** Neste vídeo, focaremos especificamente em analisar uma série de dados para que no final, possamos extrair algum conhecimento deles. Pegaremos uma série de dados misturados e seu contexto para construirmos gráficos e extrairmos médias, variante e moda, assim conseguindo entender o que eles nos dizem.
 - Desafio da aula: Fazer a lista de exercícios que foi disponibilizada e que será corrigida na última aula
- **10º vídeo - Correção dos exercícios:** Neste vídeo, estaremos corrigindo a lista de exercícios que deixamos para os alunos na última aula e estaremos tirando quaisquer dúvidas.