

Título dos encontros	Linguagens de IA		Sequência: 16
Etapa/Ano	Ensino Médio: ☒ 1º Ano ☐ 2º Ano ☐ 3º Ano		
Nível de Maturidade [link]	Escola	☐ Emergente ☒ Básico ☐ Intermediário ☐ Avançado	Docente ☒ Básico ☐ Intermediário ☐ Avançado
Competências	☐ C1 ☒ C2 ☒ C3 ☐ C4 ☐ C5		
Objetos de Conhecimento	Treinamento de um sistema de IA		
Recursos Educacionais	• Projetor • Marcadores ou canetas • Computadores com acesso à internet		
Componentes Curriculares Relacionados	☐ Arte ☐ Língua Port. ☐ Ed. Física	☐ Geografia ☐ História ☐ Matemática	☐ Língua Inglesa ☐ Ciências ☒ Computação
Palavras-chave	Treinamento, Aprendizado de Máquina, Viés		
Perguntas Importantes	• O que é uma Linguagem de Programação? • Qual o perfil do programador atual?		
Encontros	2	Total de Horas-aula	4
Objetivos	• Identificar os sucessos e limitações da IA • Conhecer o processo de aprendizado supervisionado de máquina		
Habilidades Relacionadas	BNCC: EM13CO02 - Explorar e construir a solução de problemas por meio de refinamentos, utilizando diversos níveis de abstração desde a especificação até a implementação. EM13CO04 - Reconhecer o conceito de metaprogramação como uma forma de generalização na construção de programas, permitindo que algoritmos sejam entrada ou saída para outros algoritmos. EM13CO10 - Conhecer os fundamentos da Inteligência Artificial, comparando-a com a inteligência humana, analisando suas potencialidades, riscos e limites. Referencial Curricular IA no EM: EM01IA06 - Sistemas Preditivos - Compreender como o uso de bases de dados e dicionários permite a criação de Sistemas Preditivos.		
Práticas Pedagógicas Inovadoras	☒ Aula Enriquecida com Tecnologia ☐ Ensino Híbrido: Sala de aula invertida ☐ Ensino Híbrido: rotação por estação		

[\[link\]](#)

- ☐ Ensino Híbrido: rotação individual
- ☒ Aulas Mão na massa
- ☐ Aprendizagem baseada em projetos

Sequência Didática - Encontro 1

Introdução

Antes de falarmos sobre o que é programação e linguagens de IA, temos que entender o que é o termo “Algoritmo”, que será falado ao longo de todo o plano de aula. Algoritmo é uma sequência de passos bem definidos e claramente descritos que tem um objetivo. Podemos entender um algoritmo sendo uma receita de bolo. Temos os ingredientes (entradas), a descrição de como misturamos esses ingredientes (passos e sequências lógicas) e o bolo como nosso produto final (objetivo a que o algoritmo se propõe). Após entendermos o que é um algoritmo, precisamos entender outro conceito: “Linguagem de Programação”.

Recomendamos a exibição do vídeo “Mundo dos Algoritmos - Robô Sapiens / Globo News” e “O que é Algoritmo?” (vide material complementar) para auxiliar a explicação.

Agora que compreendemos a natureza dos algoritmos como receitas para resolver problemas, podemos diferenciar claramente que eles são independentes de linguagem, existindo apenas como conceitos até serem codificados. Por outro lado, a linguagem de programação é o meio pelo qual esses conceitos abstratos se transformam em instruções concretas que um computador pode executar, atuando como a ponte entre o pensamento humano e a ação computacional.

Uma linguagem de programação é um conjunto de regras e símbolos que permite aos humanos dar instruções a um computador. De modo geral, podemos entender a linguagem de programação como sendo a língua que o computador entende. Assim como nós humanos podemos falar e compreender diferentes línguas, o computador também pode. Essas linguagens possuem um vocabulário e uma gramática específicos para que as máquinas possam compreender. Sabendo então o que algoritmos e linguagem de programação, podemos chegar aos programas, que nada mais são do que algoritmos escritos em uma linguagem de programação. Agora que entendemos todos esses conceitos, vamos iniciar falando sobre a história e a evolução da programação.

História da Programação

A história da programação e do desenvolvimento de sistemas está diretamente relacionada à evolução da tecnologia, moldando o mundo digital como o conhecemos. Os primórdios da computação foram marcados por programas escritos em linguagem de máquina, onde era preciso determinar todos os passos que cada componente do computador precisava fazer a fim de atingir o objetivo proposto. Esse período, de 1940 a 1960, foi marcado por linguagens como códigos binários e linguagens de baixo nível, onde o processo era trabalho e muito propenso a

erros e falhas. Dentre os pioneiros da computação, temos Ada Lovelace, Alan Turing, John von Neumann, entre outros, e o principal foco dos programas computacionais desenvolvidos eram para cálculos científicos e aplicações militares.

A seguir, surgiram as linguagens de alto nível, como o Fortran, Cobol e Pascal. Essas linguagens eram mais simples, facilitando o entendimento por parte do programador e, por conseguinte, seu uso. Nesse período, de 1960 a 1980, tivemos o nascimento da Engenharia de Software, onde foram estudadas e criadas abordagens sistemáticas para o desenvolvimento de programas computacionais. Nessa etapa da evolução, aplicações e sistemas surgiram para inúmeras áreas.

Entre 1980 e 2000 surgiu o conceito de Programação Orientada a Objetos, permitindo que partes dos programas pudessem ser reaproveitadas, sem a necessidade de serem criadas do zero todas as vezes que fossem necessárias. Esse período foi marcado pelo surgimento de algumas das linguagens mais utilizadas hoje, sendo elas o C++, Java e Python, assim como aplicações web e mobile.

Por fim, dos anos 2000 até o presente, temos a evolução da programação para a computação em nuvem e para a Inteligência Artificial. A computação em nuvem permite a escalabilidade dos serviços e programas, reduzindo assim os custos de desenvolvimento e manutenção dos sistemas. Nesse período pode ser observado um avanço grande no aprendizado de máquina e na automação de tarefas, junto do surgimento de linguagens de programação como o Go, criada pela Google, R, muito utilizado na análise de dados, aprendizado de máquina e estatística, além de muitas outras como TypeScript, Kotlin, Swift, entre outras.

Professor, você pode revisar com seus alunos a história da programação do site Infoescola. Recomendamos também a leitura do artigo [Evolução das Linguagens de Programação e História do desenvolvimento de software: Do início até hoje](#) (vide material complementar).

Atividade 1: Olá, mundo!

Solicite aos estudantes que pesquisem e encontrem três exemplos em diferentes linguagens de programação de como criar um programa onde a mensagem "Hello World!" é exibida na tela.

Separamos aqui alguns exemplos, porém os alunos podem trazer outras linguagens não previstas aqui:

Python:

```
print("Hello World")
```

C:

```
#include <stdio.h>
int main() {
    printf("Hello World\n");
    return 0;
}
```

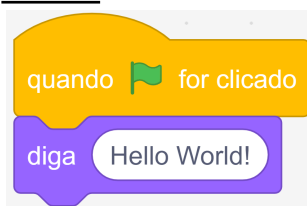
javascript:

```
<html>
<body>
  <script>
    document.write('Hello World');
  </script>
</body>
</html>
```

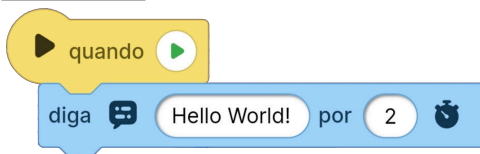
java:

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Scratch:



Octostudio:



Python: uma linguagem de programação versátil

De modo formal, podemos dizer que Python é uma linguagem de programação de alto nível, interpretada, orientada a objetos, de tipagem dinâmica e forte. Em outras palavras, Python é uma linguagem de fácil entendimento, onde podemos solicitar, de forma simples e amigável, que tarefas complexas sejam realizadas. A linguagem Python foi criada em 1991 por Guido van Rossum e ficou conhecida por sua simplicidade, legibilidade e versatilidade, tornando-a uma das linguagens mais populares do mundo, tanto para iniciantes quanto para programadores experientes. Essa característica de versatilidade permite que Python seja utilizado para desenvolvimento de sistemas Web, criação de jogos, análise de dados, automação e inteligência artificial.

Hoje é uma das linguagens mais populares no mundo. Recomendamos a exibição do vídeo "Linguagens mais populares (1965 a 2022)" onde eles poderão acompanhar a evolução das linguagens de programação mais utilizadas.

O Programador e a Programação na Era da IA

O perfil do programador de antigamente era muito diferente do que é hoje. Habilidades matemáticas eram fundamentais para lidar com os sistemas computacionais primitivos e as linguagens de baixo nível. Também era essencial que o programador fosse dotado de uma lógica impecável para escrever códigos eficientes, lidando com as limitações das máquinas da época, e livres de erros, uma vez que identificar e corrigir erros era uma tarefa complexa e difícil. Outra característica marcante dos primórdios da programação era o isolamento, uma que os programadores frequentemente trabalhavam sozinhos, sem a necessidade de interação com outras áreas.

Em contraste com o perfil inicial dos programadores, atualmente temos um programador que necessita de habilidades que vão além do conhecimento técnico acerca de linguagens de programação e algoritmos. O mercado exige um profissional cada vez mais completo, que possa aliar um sólido conhecimento técnico e capacidade de resolução de problemas com habilidade interpessoais de comunicação e trabalho em equipe.

Qualquer coisa que seja dita acerca da mudança no perfil do programador nos próximos anos é apenas uma conjectura. Apesar disso, muitas pessoas influentes no mundo corporativo apontam para uma mudança no perfil demandado.

Durante muitos anos falou-se que programação era uma habilidade essencial a ser ensinada. Porém, no último World Government Summit, que ocorreu entre os dias 12 e 14 de fevereiro de 2024 em Dubai, o Fundador e CEO da empresa NVidia, Jensen Huang, apresenta uma visão diferente (vide Blog da NVidia no material complementar):

"Na última década, quase todos os palestrantes que subiram a este palco enfatizaram que é importante que as crianças aprendam ciência da computação e saibam programar. No entanto, a situação atual é quase oposta: nosso trabalho é criar tecnologia informática para que ninguém precise programar. Agora todos, em todo o mundo, podem ser programadores. Este é o milagre da inteligência artificial. [...] Pela primeira vez, preenchemos a lacuna tecnológica."

Diferentes opiniões acerca do tema apontam para um futuro onde o perfil do programador deverá ser cada vez mais versátil e com capacidade de se adaptar às mudanças. Em especial para as mudanças promovidas pela IA, onde a interação com as ferramentas e funcionalidades será essencial em todas as áreas.

Durante os encontros de Inteligência Artificial, utilizaremos esta linguagem, pois sua sintaxe é simples e de fácil legibilidade, o que facilita o aprendizado, especialmente para iniciantes. Com uma vasta comunidade e amplas bibliotecas, oferece excelente suporte e recursos para diversas aplicações, desde desenvolvimento web até inteligência artificial.

Faça o fechamento da aula, explicando que no próximo encontro eles farão o primeiro programa em uma linguagem de programação Python.	
Avaliação	Os alunos podem ser avaliados de acordo com sua interação e participação nas discussões.
Material Complementar	Mundo dos Algoritmos - Robô Sapiens / Globo News https://youtu.be/tOYGZ9PFwLU O que é Algoritmo? https://youtu.be/e4RBMgITG44 InfoEscola - História da Programação https://www.infoescola.com/informatica/historia-da-programacao/ Evolução das Linguagens de Programação: Um Panorama Histórico até os Dias Atuais https://www.dio.me/articles/evolucao-das-linguagens-de-programacao-um-panorama-historico-ate-os-dias-atuais História do desenvolvimento de software: Do início até hoje https://awari.com.br/historia-do-desenvolvimento-de-software-do-inicio-ate-hoje/ Blog da NVidia https://blogs.nvidia.com/blog/world-governments-summit/ Linguagens mais populares (1965 a 2022) https://www.youtube.com/watch?v=qQXXI5QFUfw

Sequência Didática - Encontro 2

Introdução

Após conhecermos os conceitos teóricos acerca de linguagens de programação, vamos conhecer a linguagem Python na prática. Para isso, vamos acessar a interface de desenvolvimento chamada Google Colaboratory. Através dessa interface temos todo o ambiente e todas as ferramentas necessárias para programar em Python previamente instaladas e configuradas. Nos materiais complementares temos exemplos de algoritmos desenvolvidos para a Linguagem Python.

Orientações Gerais

Deixamos exemplos dessa atividade no material complementar. O exemplo completo pode ser encontrado em “Atividade Prática no Google Colaboratory - Completo” no material complementar.

Atividade Prática - Vamos programar em Python

Se você e seus alunos souberem programar, pode-se abrir o Google Colaboratory (vide Material Complementar) e iniciar a atividade. Contudo, se você não sabe programar, isso não é mais um problema. Atualmente não é mais necessário saber programar para criar alguns algoritmos simples. Vamos iniciar criando uma calculadora.

Essa calculadora precisa realizar as operações de adição e subtração de dois números. Para isso, vamos solicitar ao Gemini que nos ajude a criar esse algoritmo. Começamos acessando o Gemini e vamos conversar com ele, escrevendo o seguinte *prompt*:

“Estou realizando uma tarefa de programação em python e gostaria que você me ajudasse. Preciso criar um algoritmo em python que atue como uma calculadora que realiza operações de adição e subtração de dois números.”



Veja o resultado em “Atividade Prática no Google Colaboratory - Etapa 1” (material complementar)

Os resultados mostrados pelo Gemini podem variar, mas em geral ele apresentará os procedimentos de adição e subtração, além de um algoritmo que congrega as duas funções e atende o que solicitamos.

Após a criação das funções de adição e subtração, vamos pedir que ele complemente incluindo operações de multiplicação e divisão, através do *prompt*:

“Agora que temos as operações de adição e subtração, gostaria de complementar meu algoritmo com as operações de multiplicação e divisão. Por favor, gere uma versão completa do algoritmo com todas as operações e a função principal.”



Veja o resultado em “Atividade Prática no Google Colaboratory - Etapa 2” (material complementar)

Pode-se observar que o Gemini gerou o algoritmo completo apresentando as funções de adição, subtração, multiplicação e divisão. Contudo, apesar de parecer que tudo funciona adequadamente, ao tentar realizar uma divisão por zero, o algoritmo apresenta um erro e finaliza sua execução. Por mais que divisões por zero não existam (desconsiderando as questões relacionadas a número imaginários), o algoritmo deveria emitir algum alerta e informar o usuário que isso ocorreu e não deve simplesmente para sua execução. Nesse sentido, para corrigirmos o algoritmo, podemos solicitar que o Gemini faça essa correção através do *prompt*:

“Seu algoritmo apresenta erros se eu tentar dividir um número por zero. Por gentileza gere um algoritmo completo com essa função corrigida.”

 Veja o resultado em “Atividade Prática no Google Colaboratory - Etapa 3” (material complementar)

Neste exemplo, criamos um programa utilizando a linguagem Python sem necessariamente compreendermos essa linguagem. As IAs multimodais são capazes de trabalhar em diferentes domínios, com diferentes estilos de texto, criar e reconhecer imagens, montar vídeos e até mesmo programar. Dessa forma, elas não apenas são capazes de acelerar o nosso trabalho, mas permitem que utilizemos conhecimentos além daqueles que temos. Já existem estudos que afirmam que ferramentas como a Gemini e o GitHub Copilot são o equivalente a alguns anos de experiência de programação.

Agora, vamos fazer uma reflexão com os alunos, quão importante será saber programar num futuro próximo? E quanto será necessário saber conversar com as IAs sobre programação?

Atividade extra (opcional):

Proponha aos estudantes gerar outros programas simples em Python utilizando uma LLM. Algumas sugestões são: uma calculadora de média, um conversor de unidades e uma calculadora de porcentagem.

Avaliação	Os alunos podem ser avaliados de acordo com sua interação e participação no ambiente de programação.
Material Complementar	Google Colaboratory https://colab.research.google.com/ Google Gemini https://gemini.google.com/app Atividade Prática no Google Colaboratory - Etapa 1 https://colab.research.google.com/drive/1uI-CYhiaZ-9K3LOU7sTOTn4GLHSgR-AE?usp=sharing Atividade Prática no Google Colaboratory - Etapa 2 https://colab.research.google.com/drive/1IKJV8Hj4yjf3Gi1Z82pBQTH6SBGC1Z95?usp=sharing Atividade Prática no Google Colaboratory - Etapa 3 https://colab.research.google.com/drive/1_yD8ktbsv3JcShcqZF5isLC4kSvIBHKW?usp=sharing Atividade Prática no Google Colaboratory - Completo https://colab.research.google.com/drive/1RGHd2LDEVum7zewlpL8KIWy17Jc8Yt0N?usp=sharing Tutorial Python https://www.w3schools.com/python/

	Exemplo de Algoritmos em Linguagem Python - Linguagens de IA
--	---

	https://colab.research.google.com/drive/1vwMhN9sg6wATLS-j04HvmigHRxuwoBhH?usp=sharing
--	---