

TÍTULO: Encontra e corrige os erros e faz o teu programa funcionar

CENÁRIO DE APRENDIZAGEM	
Escola:	Duração (minutos): 90
Professor(a):	Idade dos alunos: 12

Ideia chave:	A maioria dos erros na escrita de um programa resume-se a sintaxe.
---------------------	--

Tópicos:

- Os alunos planeiam, antecipam, monitorizam, criam e ajustam programas.

Objetivos:

- Os alunos projetam e criam uma animação ou programa de trabalho para um propósito específico.

Resultados:

- Os alunos planeiam, antecipam, monitorizam, criam e ajustam programas.

Formas de trabalho:

- trabalho individual
- trabalho de pares
- trabalho de grupo

Métodos:

- apresentação
- discussão
- exercício interativo

ARTICULAÇÃO
Linha de atuação (duração, minutos)
<p style="text-align: center;">INTRODUÇÃO</p> <p>O Professor explica e inicia a discussão com os alunos:</p> <p>Escrever os nossos primeiros programas e colocá-los a fazer o que queremos é muito divertido, mas também é frustrante ter erros que impedem o nosso programa de trabalhar.</p> <p>Isto é especialmente frustrante se é um erro que não conseguimos encontrar rapidamente, por isso perdemos o nosso tempo que podemos estar a gastar a fazer outra coisa.</p> <p>A maioria dos erros na escrita de um programa resume-se a sintaxe – normalmente esquecemo-nos de um suporte ou de um apóstrofo, ou talvez um bloco não esteja bem ligado.</p>
<p style="text-align: center;">PARTE PRINCIPAL</p> <p><u>Tópicos para discussão</u></p> <p>Existem formas de evitar erros na escrita dos teus programas, e alguns deles são:</p> <p>- Certifica-te de que o teu código é legível.</p> <p>Não é importante a linguagem de programação estás a usar – seja em blocos em algo como Scratch ou código em algo como Python – é muito importante que o teu código possa ser facilmente lido por outras pessoas.</p> <p>E sim, isso também se aplica a programas que não vão ser vistos por mais ninguém.</p> <p>Um código facilmente legível significa que os blocos são ordenados e logicamente ordenados, ou o texto está escrito com espaçamentos claros entre linhas, quando isso é uma coisa lógica a fazer.</p> <p>Se o teu programa tem várias partes, tem sempre de ser claro qual a parte que faz o quê.</p> <p>- Utiliza comentários sempre que possível.</p> <p>Imagina que és uma pessoa que está a ver o teu código pela primeira vez. Se houver uma parte do código que não seja instantaneamente compreensível para essa pessoa (assumindo que sabe a linguagem que está a usar) – deves usar a opção de escrever comentários que expliquem o que essa parte do código está a fazer.</p> <p>- Lê os programas dos outros.</p> <p>Se queres ser um fotógrafo melhor, deves olhar para as fotografias que os profissionais fazem, e se queres tornar-te um melhor jogador de basquetebol, deves ver jogos da NBA.</p>

É sempre uma boa regra, aprender com os profissionais para se tornar melhor em qualquer área e a programação não é exceção.

Mantém-te online e vê como os outros estão a escrever o seu código na mesma linguagem de programação que estás a usar.

Se passaste por todos os passos acima e ainda está a receber uma mensagem de erro, podes tentar uma das seguintes sugestões:

- Olha para o que o teu programa te está a dizer que é o erro.

O que quer que estejas a usar para escrever o teu programa – quase sempre te dirá onde está o teu erro. Isto pode significar que marcará graficamente essa parte do programa, ou irá dizer-te em que linha de código o erro aconteceu, ou dirá qual comando ou variável com erros.

Antes de mudar qualquer outra coisa – concentra-te em mudar o que o teu programa te disser para mudares.

- Usa o Google.

Se não consegues encontrar o erro sozinho, é muito possível que seja um erro muito comum e que não sejas a primeira pessoa a tê-lo. Uma pesquisa fácil no Google pelo tipo de erro que estás a receber pode muitas vezes significar que tu vais encontrar um texto onde alguém explica como o encontrou e mais importante como o resolveu.

- Pede a outra pessoa para dar uma olhadela no teu programa.

Os programadores muitas vezes entram tão "dentro" dos seus programas que não conseguem ver a floresta em todas as árvores. Às vezes, deparamo-nos com um erro que normalmente detetamos muito facilmente, mas isso é difícil de fazer depois de teres passado horas a programar. Em momentos como este é útil pedir a outra pessoa para dar uma olhadela no nosso programa.

- Faz uma pausa.

Se nada mais está a funcionar – deixa a programação, mas apenas por um tempo. Afasta-te do computador, come algo, assiste a um episódio do teu programa favorito, vai lá fora dar uma volta ou vai dormir uma sesta.

O erro ainda estará lá quando voltares, mas o teu cérebro vai estar a trabalhar no fundo à procura de uma solução e é muito possível que venhas a ter uma ideia que ainda não tentaste.

Exercício 1

O professor explica e dá instruções sobre como resolver tarefas.

Os alunos criam uma simples animação no Scratch.

Aplicam conhecimentos e competências anteriormente adquiridos.
Os alunos no seu trabalho criam deliberadamente um erro como armadilha.
Os alunos partilham os seus trabalhos e descobrem erros nos programas de outros alunos.
Os alunos resolvem tarefas e apresentam as suas soluções.

CONCLUSÃO

Os alunos e o professor discutem e avaliam as soluções apresentadas.

Métodos

apresentação
discussão
trabalhar no texto
trabalho gráfico
exercício interativo / simulação no computador

entrevista
demonstração
representação

Formas de trabalho

Trabalho individual
Trabalho em pares
trabalho em equipa/grupo
trabalho frontal

Material:

-

Bibliografia:

OBSERVAÇÕES PESSOAIS, COMENTÁRIOS E NOTAS