

Plano de Aula – Crash e Depuração: Corrigindo Erros com Programação

Tema

Identificação e correção de erros em projetos de programação com o Robô VEX 123, desenvolvendo o raciocínio lógico e o pensamento computacional.

Duração da Aula

1 aula de 50 minutos.

Componentes Curriculares Envolvidos

- Matemática (lógica e sequência)
 - Tecnologias e Robótica Educacional
 - Educação Socioemocional (paciência e persistência)
 - Ciências (exploração e análise)
-

Turmas Indicadas

- 2º ao 5º ano do Ensino Fundamental
-

Objetivos da Aula

- Compreender o conceito de depuração (debug) em programação.
 - Desenvolver a habilidade de testar, identificar e corrigir erros.
 - Estimular a análise crítica de trajetos e ações programadas.
 - Trabalhar o planejamento e revisão de projetos.
 - Promover o trabalho em equipe na busca por soluções.
-

Competências da BNCC

- Competência Geral 1: Utilizar conhecimentos para resolver situações-problema.
 - Competência Geral 4: Utilizar diferentes linguagens, incluindo a computacional.
 - Competência Geral 5: Compreender e utilizar tecnologias digitais.
 - Competência Geral 6: Raciocinar logicamente com base em hipóteses e testes.
-

Habilidades da BNCC

- (EF02ET02) Usar comandos de programação para resolver problemas.
 - (EF15MA01) Formular e testar hipóteses para resolver desafios.
 - (EF02CI01) Observar e investigar situações para buscar soluções.
 - (EF02MA10) Reconhecer padrões de repetição e sequência.
-

Materiais Necessários

- Robôs VEX 123 (1 por grupo)
 - Campo 123 ou área plana com ladrilhos
 - Pompons ou objetos pequenos para representar obstáculos
 - Tablets ou computadores com VEXcode 123
 - Papel para anotações e planejamento
-

☐ Etapas e Desenvolvimento da Aula (Passo a Passo)

1. Introdução ao Conceito (10 min)

- Explique o que significa depurar: encontrar e corrigir erros no código.
- Mostre que erros fazem parte do processo de aprendizagem.
- Apresente o desafio: o robô deve atravessar o campo sem tocar no obstáculo central VEX 123 - Crash e depur....

2. Primeiro Teste com Código (10 min)

- Os alunos usam um código fornecido (pré-definido ou com erros intencionais).
- Colocam o robô no campo, com um pompom no centro, e observam se ele evita o obstáculo.
- Analisam o resultado e anotam o que deu errado.

3. Processo de Depuração (10 min)

- Cada grupo discute quais blocos devem ser adicionados, removidos ou reorganizados.
- Ajustam o código para evitar o obstáculo e testam novamente.
- Repetem esse ciclo até o robô chegar ao outro lado do campo sem colidir.

4. Variações e Testes (10 min)

- Mudam a posição do pompom e aplicam o mesmo código para verificar se ainda funciona.
- Com dois obstáculos, programam o robô para se desviar de ambos.

5. Discussão e Registro (10 min)

- Cada grupo relata as mudanças feitas no código e o que funcionou.
 - Refletem sobre a importância dos testes e da paciência no processo de programação.
-



Subindo de Nível

- Criar um labirinto simples com múltiplos obstáculos.
 - Usar sensores (se disponíveis) para desviar automaticamente.
 - Trabalhar com blocos condicionais: “se houver obstáculo, vire”.
-



Conteúdos Trabalhados

- Lógica e sequência de comandos
 - Testes, hipóteses e verificação
 - Pensamento computacional
 - Colaboração e análise crítica
 - Ajustes e melhoria contínua
-



Dicas para o Professor

- Mostre um exemplo em que o robô falha e resolvam juntos.
 - Valorize a tentativa e erro como parte do aprendizado.
 - Oriente para que cada ajuste seja pequeno e justificado, evitando mudanças aleatórias.
-



Discussões e Conclusões

- Qual foi o principal erro no código inicial?
 - O que vocês fizeram para corrigir?
 - Como se sentiram ao ver o robô acertar o trajeto depois de corrigir?
-



Interdisciplinaridade

- **Matemática:** lógica e análise de sequência.
 - **Tecnologia:** linguagem de programação.
 - **Ciências:** observação e análise experimental.
 - **Educação Socioemocional:** perseverança, cooperação e foco.
-



Avaliação Formativa

- Participação no processo de análise e correção.
 - Clareza na justificativa dos ajustes.
 - Compreensão do efeito de cada comando.
 - Colaboração no grupo para resolver o desafio.
-



Dicas Pedagógicas

- Use termos como “engenheiro de solução” para motivar os alunos.
 - Crie um “caderno de depuração” com erros comuns e como corrigi-los.
 - Estimule a verbalização do raciocínio antes de alterar o código.
-



Resultados Esperados

- Maior autonomia na correção de erros de programação.
- Desenvolvimento do raciocínio lógico e da análise crítica.
- Compreensão de que errar é parte do processo criativo.
- Aprendizado mais profundo por meio da prática e da persistência.