

Planificação anual

Disciplina: Pensamento Computacional (OFC) Ano: 6º Ano letivo 2025/2026

Temas/Conteúdos	N.º de aulas previstas
Pensamento Computacional (Competências a Desenvolver)	
Análise Identificar e compreender os diferentes elementos de um problema.	Todo o ano letivo
Abstração Identificar o que é importante e remover detalhes desnecessários.	
Decomposição Dividir um problema em etapas menos complexas para reduzir a sua dificuldade. Identificar todas as etapas na resolução do problema.	
Reconhecimento de Padrões Identificar semelhanças ou características entre problemas e resolver o problema utilizando soluções previamente definidas noutros problemas, e baseadas em experiências anteriores.	
Pensamento Algorítmico Ordenar sequencialmente um conjunto de instruções / etapas para resolver o problema.	
Raciocínio Lógico <i>Dedução</i> - concluir algo a partir de informações que já existem. <i>Indução</i> - afirmar uma verdade generalizada a partir da observação de alguns elementos. <i>Abdução</i> - procurar novas ideias e conhecimentos que possam validar algo.	
Avaliação Garantir que a solução é adequada para resolver o problema, verificando os casos limite.	
Pensamento Computacional (Atitudes a estimular)	
Colaboração Trabalhar em equipa.	
Criação Projetar e construir/produzir com novidade.	

Temas/Conteúdos	N.º de aulas previstas
Depuração Encontrar e corrigir erros.	
Experimentação Alterar e ver o que acontece, experimentos de causa e efeito.	
Perseverança Continuar e nunca desistir, mesmo quando o problema é mais complexo de resolver.	
Planeamento Definir objetivos de longo prazo e os meios para alcançá-los.	
Otimização Reestruturar para uma solução mais eficiente.	
Adaptação Ajustar a estratégia/resolução a cada situação.	
Conceitos Transversais	
Problema Examinar o enunciado de Problemas simples, do dia a dia, ou pequenas charadas. Identificar as informações relevantes e o objetivo da resolução.	
Dados Reforçar o conceito de Dados como expressão factual de uma propriedade de algo, alguma coisa (uma medida, um identificador, etc.).	
Tipo de Dados Identificar diferentes tipos de dados e a sua utilidade na resolução de problemas.	
Criar e Inovar	
Computador Aplicar o computador como ferramenta para resolver problemas e executar instruções programadas.	
Algoritmo Reforçar o conceito de algoritmo como sequência de passos. Examinar algoritmos simples utilizados em tarefas do quotidiano. Criar algoritmos complexidade média para a resolver desafios e problemas específicos. Otimizar algoritmos testando e ajustando etapas para maior eficiência.	

Temas/Conteúdos	N.º de aulas previstas
Instrução Reforçar o conceito de instrução como uma ordem ou comando claro dado ao computador.	
Linguagem Visual Reforçar o uso de Linguagens Visuais, diagramáticas ou icónicas, para construir algoritmos simples.	
Ambiente de Programação Usar ambientes gráficos de programação por blocos para criar e testar programas simples.	
Programa Reforçar o conceito de programa como uma sequência de instruções organizada e executável. Criar programas simples de computador para automatizar pequenas tarefas do dia a dia ou ajudar a resolver charadas. Examinar programas já feitos, compreendendo a lógica por trás da sua estrutura. Executar programas em ambiente de programação, observando o seu funcionamento. Modificar programas existentes, ajustando blocos e instruções para novos objetivos. Otimizar programas com base em testes e correções.	
Código Reforçar o conceito de Código, como um identificador único de uma entidade ou objeto.	
Bloco de código Aplicar blocos de código em linguagens visuais para construir sequências lógicas.	
Constante Aplicar o conceito de constante como um valor fixo durante a execução de um programa.	
Variável Reforçar o conceito de variável como um elemento que armazena valores alteráveis num programa.	
Sensor Aplicar sensores em projetos simples, reconhecendo como recolhem dados do ambiente (ex: luz, som, movimento) e os utilizam na programação de respostas automatizadas.	
Estrutura de Controlo Reforçar a compreensão de estruturas de controlo como ciclos (repetição) e decisões (condicionais).	
Operação Aritmética Reforçar a utilização de operações como soma, subtração, multiplicação e divisão em contextos computacionais.	

Temas/Conteúdos	N.º de aulas previstas
Operação Relacional Reforçar o uso de operadores relacionais (>, <, =) para comparar dados.	
Operação Booleana Conhecer operadores lógicos (E, OU, NÃO) e o seu papel nas decisões dos algoritmos.	
Erro Corrigir erros de lógica e de sintaxe identificados nos programas, usando a depuração. Reforçar a capacidade de identificar diferentes tipos de erros durante a execução de programas, como bugs ou falhas de lógica.	
Placas Eletrónicas com Sensores Conhecer placas eletrónicas com sensores integrados (como temperatura, luz, inclinação, botão, etc.), programando respostas automáticas a estímulos do ambiente. Usar o Micro:bit em projetos que envolvam algoritmos, estruturas de controlo e variáveis, promovendo a experimentação prática.	
Robot Reforçar o conceito de Robot e o seu funcionamento básico como sistema programável que interage com o meio. Montar Robots simples, identificando os seus componentes principais (sensores, motores, estrutura física, etc.). Usar Robots para, através de algoritmos, realizar tarefas que resolvem problemas simples.	