

CURSO DE LOGICA DE PROGRAMACAO E ALGORITMOS

Nivel Inicial

Duracao: 20 horas	Organizacao: 10 sessoes x 2 horas	Nivel: Iniciacao
Metodologia: Aprender fazendo	Projeto final: Quiz / miniaplicacao	Pre-requisitos: Nao exige programacao previa

Finalidade do curso

Desenvolver o raciocinio logico e a capacidade de resolver problemas atraves de algoritmos, fluxogramas, pseudocodigo e programacao visual. O curso privilegia exemplos do quotidiano e desafios progressivos, preparando o formando para cursos posteriores de Scratch, Python ou outras linguagens.

Programa de formacao - versao desenvolvida

1. Enquadramento, objetivos e competencias

Enquadramento

A logica de programacao e a base de qualquer linguagem informatica. Antes de aprender sintaxe, e fundamental saber decompor um problema, identificar dados, definir regras, tomar decisoes e repetir tarefas de forma estruturada. Este curso foi concebido para iniciantes e utiliza linguagem simples, exemplos visuais e exercicios orientados.

Objetivos gerais

- Compreender o conceito de algoritmo e reconhecer algoritmos no dia a dia.
- Decompor um problema em etapas simples, ordenadas e verificaveis.
- Distinguir dados de entrada, processamento e dados de saida.
- Utilizar variaveis, constantes e tipos de dados elementares.
- Aplicar operadores aritmeticos, relacionais e logicos.
- Construir estruturas condicionais simples e compostas.
- Utilizar estruturas de repeticao e contadores.
- Representar algoritmos atraves de fluxogramas e pseudocodigo.
- Testar algoritmos, localizar erros e melhorar solucoes.
- Aplicar os conhecimentos num pequeno projeto final.

Competencias a desenvolver

Raciocinio	Pensamento logico, sequenciacao, abstracao e decomposicao de problemas.
Algoritmos	Leitura, escrita, interpretacao e teste de algoritmos simples.
Representacao	Uso de fluxogramas, pseudocodigo e programacao visual.
Resolucao de problemas	Comparar solucoes, detetar erros e selecionar estrategias adequadas.

Principio metodologico

Cada novo conceito e apresentado em quatro etapas: compreender -> observar um exemplo -> resolver um desafio -> testar e corrigir.

2. Estrutura do curso - 10 sessões

Sessão	Tema	Conteúdos essenciais	Produto / desafio
1	Pensamento lógico e algoritmos	Problema, solução, instrução, sequência, início/fim.	Algoritmo de uma tarefa quotidiana.
2	Sequências e decomposição	Dividir problemas; passos; ordem; clareza; casos simples.	Algoritmo para organizar uma rotina escolar.
3	Entrada, processamento e saída	Dados, operações, resultados; leitura e escrita de dados.	Cálculo de média e preço final.
4	Variáveis e tipos de dados	Variáveis, constantes, números, texto, booleanos, atribuição.	Ficha digital de aluno.
5	Operadores	Aritméticos, relacionais e lógicos; precedência simples.	Calculadora lógica / comparações.
6	Decisões	SE, SENÃO, condições simples e compostas.	Aprovado/reprovado; classificação por idade.
7	Repetições	Ciclos, contador, acumulador, PARA, ENQUANTO.	Tabuada e contagem automática.
8	Fluxogramas	Símbolos, ligações, decisão, processo, entrada/saída.	Fluxograma completo de um problema.
9	Pseudocódigo e programação visual	Conversão de fluxograma; eventos e blocos; teste.	Mini projeto em Scratch ou equivalente.
10	Projeto final e apresentação	Planeamento, construção, teste, correção, apresentação.	Quiz escolar ou miniaplicação lógica.

Distribuição recomendada de cada sessão: 20-25 min de explicação e demonstração; 65-75 min de prática orientada; 15-20 min para desafio, correção e síntese.

3. Desenvolvimento das sessões 1 a 5

Sessão 1 - Pensamento lógico e algoritmos

- O que é um problema e o que significa encontrar uma solução.
- Conceito de algoritmo: conjunto finito e ordenado de instruções.
- Importância da ordem, clareza e ausência de ambiguidades.
- Início, desenvolvimento e fim de uma tarefa.

Exemplo: algoritmo para iniciar um computador e abrir um documento.

Exercício: escrever um algoritmo com 6 a 10 passos para preparar a mochila escolar. Trocar com um colega e testar se as instruções são suficientemente claras.

Sessão 2 - Sequências e decomposição

- Decompor tarefas complexas em pequenas tarefas.
- Sequências lineares e dependência entre passos.
- Detetar passos em falta, repetidos ou fora de ordem.
- Introdução ao conceito de caso de teste.

Exemplo: decompor 'entregar um trabalho na plataforma escolar' em subtarefas.

Exercício: ordenar um conjunto de instruções baralhadas e justificar a ordem final.

Sessão 3 - Entrada, processamento e saída

- Identificar dados de entrada e resultados esperados.
- Processamento: cálculos, transformações e comparações.
- Leitura e apresentação de dados.
- Modelo Entrada -> Processamento -> Saída.

Exemplo: receber três classificações, calcular a média e apresentar o resultado.

Exercício: criar o algoritmo para calcular o preço final de um produto com 10% de desconto.

Sessão 4 - Variáveis e tipos de dados

- Conceito de variável e atribuição.
- Constantes e valores que não mudam.
- Tipos elementares: inteiro, real, texto e booleano.
- Nomes claros para variáveis.

Exemplo: nome='Ana', idade=12, nota=16, aprovado=verdadeiro.

Exercício: criar as variáveis necessárias para representar uma ficha de aluno e indicar o tipo de cada uma.

Sessão 5 - Operadores

- Operadores aritméticos: +, -, *, / e resto da divisão.
- Operadores relacionais: >, <, >=, <=, = e diferente.
- Operadores lógicos: E, OU e NÃO.
- Expressões simples e interpretação de resultados.

Exemplo: (idade >= 12) E (autorização = verdadeiro).

Exercício: construir expressões para verificar idade, classificação, preço e acesso a uma atividade.

4. Desenvolvimento das sessões 6 a 10

Sessão 6 - Estruturas de decisão

- Condição e resultado verdadeiro/falso.
- Estrutura SE ... ENTÃO ... SENÃO.
- Condições encadeadas e condições compostas.
- Escolha do caminho adequado consoante os dados.

Exemplo: se média ≥ 10 , apresentar 'Aprovado'; caso contrário, 'Não aprovado'.

Exercício: classificar uma temperatura como baixa, moderada ou elevada com regras previamente definidas.

Sessão 7 - Estruturas de repetição

- Por que motivo repetimos instruções em programação.
- Contador e acumulador.
- Ciclo PARA em tarefas com número conhecido de repetições.
- Ciclo ENQUANTO em tarefas dependentes de uma condição.

Exemplo: apresentar os números de 1 a 10 e construir uma tabuada.

Exercício: somar cinco valores introduzidos e apresentar o total e a média.

Sessão 8 - Fluxogramas

- Símbolo de início/fim, processo, decisão e entrada/saída.
- Fluxo de execução e setas.
- Como representar condições e repetições.
- Leitura e validação de um fluxograma.

Exemplo: fluxograma para verificar se um número é positivo, negativo ou zero.

Exercício: construir o fluxograma do algoritmo de aprovação escolar criado na sessão 6.

Sessão 9 - Pseudocódigo e programação visual

- Regras simples de pseudocódigo.
- Conversão de fluxograma para pseudocódigo.
- Eventos, blocos, variáveis e condições em Scratch ou ferramenta equivalente.
- Testes e depuração elementar.

Exemplo: transformar um algoritmo de quiz em blocos de programação.

Exercício: programar 3 perguntas com pontuação e mensagem final.

Sessão 10 - Projeto final

- Escolha e planeamento do problema.
- Definição de entradas, regras e saídas.
- Fluxograma e pseudocódigo.
- Implementação, testes, correção e apresentação.

Projeto recomendado: Quiz Escolar, Calculadora de Média, Jogo de Adivinhação ou Classificador de Idade.

Entrega: algoritmo + fluxograma + implementação visual + breve apresentação oral.

5. Banco de exercicios complementares

Exercicio A - Sequenciacao

Ordena 8 passos baralhados para efetuar uma pesquisa segura na Internet. Identifica um passo que poderia ser eliminado e justifica.

Exercicio B - Detetar o erro

E fornecido um algoritmo com um passo em falta. O formando deve identificar o erro, corrigir a sequencia e testar a nova versao.

Exercicio C - Media escolar

Ler quatro notas, calcular a media e apresentar 'Aprovado' se a media for igual ou superior a 10.

Exercicio D - Par ou impar

Ler um numero inteiro e determinar se e par ou impar usando o resto da divisao.

Exercicio E - Maior de tres

Ler tres numeros e indicar qual e o maior. Discutir o que acontece se existirem valores iguais.

Exercicio F - Tabuada

Pedir um numero e apresentar automaticamente a respetiva tabuada de 1 a 10.

Exercicio G - Contagem regressiva

Apresentar 10, 9, 8 ... 1 e terminar com uma mensagem final.

Exercicio H - Login simples

Comparar nome de utilizador e palavra-passe com valores definidos e apresentar acesso permitido ou negado.

Exercicio I - Classificacao

Converter uma nota numerica numa categoria qualitativa segundo regras fornecidas pelo formador.

Exercicio J - Quiz

Criar 5 perguntas, atribuir 1 ponto a cada resposta correta e apresentar o total no final.

6. Metodologia, avaliacao e recursos

Metodologia de formacao

- Exposicao curta com exemplos concretos e linguagem acessivel.
- Demonstracao visual de algoritmos, fluxogramas e blocos de programacao.
- Exercicios progressivos: primeiro guiados, depois semi-orientados e finalmente autonomos.
- Trabalho por pares para comparar solucoes e discutir diferentes algoritmos.
- Testes com dados diferentes para verificar se a solucao funciona em mais de um caso.
- Correcao comentada, incentivando o formando a explicar o seu raciocinio.

Avaliacao sugerida

Componente	Peso	Indicadores
Exercicios de aula	30%	Participacao, aplicacao dos conceitos, autonomia progressiva.
Desafios praticos	25%	Correcao logica, clareza e capacidade de teste.
Fluxograma e pseudocodigo	20%	Estrutura, simbolos, sequencia e coerencia.
Projeto final	25%	Planeamento, funcionamento, testes e apresentacao.

Recursos

- Computador por formando ou por par.
- Projetor / ecrã para demonstracoes.
- Editor de texto simples para pseudocodigo.
- Ferramenta de fluxogramas (por exemplo, diagrams.net ou equivalente).
- Scratch ou plataforma de programacao visual equivalente.
- Fichas de exercicios e grelhas de avaliacao.

Nota ao formador

O foco deve permanecer na logica e nao na memorizacao de sintaxe. Sempre que possivel, peça ao formando para explicar oralmente o algoritmo antes de o representar ou programar.

7. Projeto final - Miniaplicacao logica

Objetivo

Consolidar os conceitos do curso atraves da resolucao completa de um pequeno problema, desde a ideia inicial ate ao teste da solucao.

1. Definir o problema	Explicar em uma ou duas frases o que a aplicacao deve fazer.
2. Identificar dados	Listar entradas, variaveis, regras e saidas.
3. Criar o algoritmo	Escrever a sequencia de passos em linguagem simples.
4. Representar	Construir um fluxograma e uma versao em pseudocodigo.
5. Implementar	Criar a versao em Scratch ou ferramenta visual equivalente.
6. Testar	Usar pelo menos tres conjuntos de dados ou situacoes diferentes.
7. Corrigir	Registrar erros encontrados e as alteracoes efetuadas.
8. Apresentar	Demonstrar o funcionamento e explicar a logica utilizada.

Sugestoes de projeto

- Quiz escolar com 5 a 10 perguntas e sistema de pontuacao.
- Calculadora de media com classificacao final.
- Jogo de adivinhacao de numero.
- Assistente simples para escolher uma atividade com base em idade e preferencias.
- Conversor simples (por exemplo, minutos para horas/minutos).

Criterio de sucesso

O projeto nao precisa de ser complexo. Deve, acima de tudo, demonstrar que o formando consegue definir um problema, criar uma sequencia logica, usar pelo menos uma decisao ou repeticao e testar o resultado.