

## Plano de Ensino

**Disciplina:** Desenvolvimento Desktop

**Período:** 1º Semestre

**Curso:** Análise e Desenvolvimento de Sistemas

**Carga Horária:** 80 horas

**Docente:** Fabiane Sorbar

**Titulação:** Mestrado

### Ementa

Conceitos básicos na construção de algoritmos (lógica, variáveis, constante, operadores aritméticos, relacionais e lógico) e estruturas de controle (sequencial, condicional e de repetição). Variáveis compostas homogêneas e heterogêneas. Arquivos. Modularização de algoritmos: procedimentos e funções. Estruturação e depuração de programas em linguagem de alto nível. Resolução de problemas aplicados a uma situação real implementada em um software desktop.

### Objetivo Geral

Ser capaz de estruturar soluções para problemas em níveis distintos de abstrações formalizados em linguagem computacional.

### Objetivo Específico

Saber utilizar o raciocínio lógico na formalização de soluções;  
Ser capaz de, em diversos níveis de abstração, identificar os passos necessários para a solução de problemas;  
Compreender a sintaxe e semântica de uma linguagem de programação para o desenvolvimento de aplicações.

### Conteúdo Programático

1. Aula expositiva apresentando o professor, o projeto e como ele será desenvolvido.  
Dinâmica: Iniciar a aula pedindo para os alunos apresentarem sua compreensão sobre o que é um computador e o que esses dispositivos são capazes de realizar e suas limitações. O que é um algoritmo e a relação Hardware x Software. Analisar as respostas e mostrar as capacidades de um computador na resolução de problemas.  
Gamificação como atividade para interação.
2. Apresentar os conceitos iniciais sobre a Linguagem de Programação, tipos de variáveis, operadores matemáticos utilizados, relação hardware x software na linguagem.  
Atividade de integração com a simulação de resolução de problemas do dia a dia, selecionando variáveis e cálculos envolvidos.
3. Explorar os comando de Entrada e Saída.  
Escolha do tema para projeto.  
Colocar em prática as atividades teóricas realizadas da aula anterior, dentro do software de programação.
4. Instruções de Seleção - if - else.  
Resolução de Estudos de Caso sobre o conteúdo abordado.
5. Instruções de Seleção - Switch  
Resolução de Estudos de Caso sobre o conteúdo abordado.  
Aprendizagem entre pares e times - If / Else ou Switch?
6. Instruções de Repetição - While (Análise início e fim).  
Resolução de Estudos de Caso sobre o conteúdo abordado.
7. Instruções de Repetição - For  
Resolução de Estudos de Caso sobre o conteúdo abordado.  
Aprendizagem entre pares e times - While ou For?
8. Entrega parcial do projeto, contendo parte inicial implementada até o momento.
9. Prova Dissertativa, individual, avaliando conteúdos abordados até a presente data.
10. Estudo Dirigido Realização de Unidade de Aprendizagem sobre Funções e Procedimentos.
11. Apresentar os conceitos sobre funções secundárias, criação de arquivos complementares e projetos.  
Metodologia ativa Cultura Maker para direcionamento e distribuição das ideias sobre os assuntos e resolução das atividades.
12. Apresentar Vetores e Matrizes e suas formas de aplicação.  
Metodologia ativa através da Gamificação, onde eles construirão seus próprios jogos com a utilização dos recursos aprendidos em sala.
13. Semana Acadêmica ADS e C&T
14. Apresentar sobre estruturas em C, um tipo de variável criada pela programação, que pode ser associada a vetores e matrizes. Estudo dirigido com resolução de problemas sobre o assunto.  
Discutir também, sobre os Métodos de Ordenação.  
Sala de Aula Invertida.

<b>Disciplina:</b> Desenvolvimento Desktop	<b>Período:</b> 1º Semestre
<b>Curso:</b> Análise e Desenvolvimento de Sistemas	<b>Carga Horária:</b> 80 horas
<b>Docente:</b> Fabiane Sorbar	<b>Titulação:</b> Mestrado

15. O tipo de dado ponteiro  
Passagem de parâmetro por referencia.
16. Alocação Dinâmica x Estática  
Princípios da alocação dinâmica para variáveis simples e compostas.
17. Ensinar como criar arquivos texto a partir de programas.  
Estudos de Caso utilizando problemas do dia a dia aplicados a situação.
18. Ensinar a criar e manipular arquivos binários.
19. Avaliação objetiva
20. Apresentação Projeto Final disciplina

## Metodologia

A metodologia Biopark Educação enfatiza a aprendizagem efetiva, atribuindo significado ao processo educativo. A metodologia de ensino e aprendizagem adotada é prática, experimental e de projetos, nos quais os alunos resolvem problemas reais da comunidade, inovam na criação de produtos e são estimulados para a pesquisa. Os pilares da metodologia Biopark Educação valorizam elementos que desenvolvem competências pessoais e profissionais. Requerem-se profissionais capacitados para atuar no mundo do trabalho, com responsabilidade ética e valores humanos. Os alunos do Biopark Educação compartilham experiências inovadoras em educação, permeadas pelo uso de tecnologias, aprender fazendo, desenvolvimento da criticidade e solidariedade, em prol da construção e transformação da sociedade em lugar melhor de viver. Vivenciam a Sala de Aula Invertida, na qual fazem o estudo prévio dos conteúdos, sob forma de Estudo Independente, para depois, em sala, com auxílio dos colegas e docentes, dar aplicabilidade aos conhecimentos. Em sala de aula, o docente faz a mediação do processo de aprendizagem, fazendo uma breve contextualização da temática em estudo, explicando quando for o caso (individual ou coletivamente), coordenando a discussão, fazendo a síntese e o aprofundamento. É contínuo a prática de solucionar problemas e desafios. Esta é a forma inovadora de se fazer ensino superior, e é a forma de possibilitar que o aluno se torne protagonista de sua formação, envolvendo-o o tempo todo em atividades dinâmicas e interessantes.

## Sistema de Avaliação

O processo avaliativo do Biopark Educação orienta-se pela Avaliação Formativa Mediadora, na qual o estudante é orientado a avançar na sua aprendizagem, respeitando-se o tempo de aprender. A avaliação ocorre ao longo do processo educativo, no qual o estudante aprende fazendo e o docente orienta e média as etapas de aprendizagem.

A composição das notas é pela Rubrica CHA: Conhecimento, Habilidade e Atitudes

Rubrica CHA: Conhecimento, Habilidade e Atitudes sendo:

Conhecimentos 70%

- . Avaliação objetiva 15%
- . Unidades de Aprendizagem (Uas) - 10%
- . Avaliação dissertativa 30%
- . Entrega 15%

Habilidades e Atitudes - 30% (Avaliação 360°)

Habilidades

- . Comunicação Oral e Escrita - 10%
- . Cognitivo - 10%

Atitudes

- . Autogestão - 2,5%
- . Autonomia - 2,5%
- . Protagonismo 2,5%
- . Interação - 2,5%

A avaliação 360° será organizada da seguinte maneira:

- . 3% Autoavaliação
- . 3% Equipe (todos integrantes da equipe avaliam o colega)
- . 24% Professor

## Bibliografia Básica

FORBELLONE, André Luiz Vilar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados . 3. São Paulo: Prentice Hall, 2005, 218 p.

OLIVEIRA, Cláudio Luís Vieira; LUHMANN, Ângela Cristina de Oliveira. Aprenda lógica de programação e algoritmos: com implementações em Portugol, Scratch, C, Java, C# e Python. Rio de Janeiro: Editora Ciência Moderna, 2016, 339 . p.

MANZANO, José Augusto N.G. Algoritmos: lógica para desenvolvimento de programação de computadores. 29.. São Paulo: Editora Érica Ltda., 2019. 368 p.

**Disciplina:** Desenvolvimento Desktop

**Período:** 1º Semestre

**Curso:** Análise e Desenvolvimento de Sistemas

**Carga Horária:** 80 horas

**Docente:** Fabiane Sorbar

**Titulação:** Mestrado

## Bibliografia Complementar

SEBESTA, Robert W.; TORTELLO, João Eduardo Nobrega (Trad.). Conceitos de linguagem de programação. 11. Porto Alegre: Bookman, 2018, 757 p.

FARRER, H.; BECKER, C. G.; FARIA, E. C.; MATOS, H. F.; SANTOS, M. A.; MAIA, M. L. Programação estruturada de computadores: algoritmos estruturados. 3. Rio de Janeiro: Livros Técnicos e Científicos Editora S.A.-LTC, 2017.

SOUZA, M A F. GOMES, M M. SOARES, M V. CONCILIO, R. Algoritmos e lógica de programação: um texto introdutório para a engenharia. 3. São Paulo: Cengage Learning, 2020, xxvii, 272 p.

CELES, Waldemar; RANGEL, José Lucas; CERQUEIRA, Renato. Introdução a estruturas de dados: com técnicas de programação em C. 2. v. Rio de Janeiro: Editora Elsevier Ltda, 2016, 394 p.

MIZRAHI, Victorine Viviane. Treinamento em linguagem C. 2. São Paulo: Pearson/Prentice Hall, 2008, 405 p.

Revista de Ciências da Computação. Universidade Aberta. Disponível em: <https://journals.uab.pt/index.php/rcc/index>

## TDE (Trabalho Discente Efetivo)

TDE:

As atividades do TDE foram realizadas pelos estudantes fora do horário de aulas, em forma de estudo independente, antes ou após as atividades de sala de aula.

As atividades de TDE consistiram em: implementação de códigos, teste em códigos, leitura de textos, artigos, análise de casos, elaboração de sínteses e resumos.

As atividades de TDE foram acompanhadas pela professor e as suas entregas, estão previstas no cronograma da disciplina e somam com as notas das demais avaliações, totalizando 14 horas.

Coordenadora: Fabiane Sorbar

  
Diretora Acadêmica: Dayane Kelly Sabec Pereira