

**PLANO DE AULA****DISCIPLINA:** Programação Imperativa e Funcional -Turma B**CÓDIGO:****PROFESSOR:** Pamela Thays Lins Bezerra**CARGA HORÁRIA:** 60 horas**CURSO:** Ciência da Computação**PERÍODO:** 2021.2**EMENTA:**

Conceituação e introdução a paradigmas de programação. Introdução ao paradigma imperativo através de programação em C: comandos de entrada e saída; estruturas de decisão e repetição; funções, matrizes, strings e tipos de dados estruturados; subprogramas, arquivos, conjuntos; ponteiros e alocação de memória. Introdução ao paradigma funcional através de programação em Haskell: listas, tuplas, tipos, classes, polimorfismo, funções de alta ordem, e recursão. Resolução de problemas em computação usando ambas as linguagens.

OBJETIVOS:

A disciplina Programação Imperativa e Funcional tem por objetivo introduzir conceitos de paradigmas de programação e explorar outras linguagens de programação além daquelas estudadas em Fundamentos de Programação no primeiro período. Essa introdução permite, portanto, aos alunos expandirem seu conhecimento sobre programação, lógica computacional, gerenciamento de memória, e desenvolvimento de software, habilidades técnicas essenciais para qualquer cientista da computação.

Por objetivos específicos temos:

- Introduzir o conceito de paradigmas de programação;
- Introduzir a linguagem de programação C;
- Estudar os conceitos básicos para desenvolvimento de projetos em C: comandos de entrada e saída; estruturas de decisão e repetição; funções, matrizes, strings e tipos de dados estruturados; subprogramas, arquivos, conjuntos; ponteiros e alocação de memória;
- Apresentação ao paradigma funcional e a linguagem de programação Haskell;
- Explorar os conceitos básicos para o desenvolvimento de projeto em Haskell: listas, tuplas, tipos, classes, polimorfismo, funções de alta ordem, e recursão;
- Resolução de problemas utilizando ambas as linguagens;
- Técnicas de desenvolvimento de projetos em ambas as linguagens;

PROGRAMA:

ENC ONT RO	DATA	Carga Horária em Minutos		CONTEÚDO	ESTRATÉGIA	RECURSOS	ESPAÇO
		Teórica	Prática				
1	17/08/2021	90	0	Introdução a Programação	Apresentação do curso, os conceitos básicos de paradigmas de programação, introdução a C e a Haskell, e discussão sobre as regras da CESAR School	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
2	20/08/2021	90	0	Estruturas de Decisão e Repetição	Aula Teórica – explorar os blocos de decisão(if-then-else) e repetição (for e while loops)	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)

3	23/08/2021	0	60	Acompanhamento de Projeto	Acompanhamento individual de projeto e esclarecimento de dúvidas	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
4	24/08/2021	90	0	Matrizes e funções	Aula Teórica – Descrever como implementar uma matriz em C. Explicar como desenvolver funções em C e tornar seu código mais modularizado.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
5	27/08/2021	90	0	Funções e Structs	Aula Teórica – Continuação da explicação sobre funções com exemplos. Introdução a structs.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
6	31/08/2021	90	0	KICK-OFF	-----	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
7	03/09/2021	90	0	Structs e Ponteiros	Aula Teórica – Continuação da explicação de structs com exemplos. Introdução ao conceito de ponteiros.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
8	06/09/2021	0	0	FERIADO	-----	-----	-----
9	07/09/2021	0	0	FERIADO	-----	-----	-----
10	10/09/2021	90	0	Ponteiros e Alocação Dinâmica de memória	Aula Teórica – Continuação de ponteiros com exemplos. Introdução a alocação dinâmica de memória.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
11	13/09/2021	0	60	Acompanhamento de Projeto	Acompanhamento individual de projeto e esclarecimento de dúvidas	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
12	14/09/2021	90	0	Alocação dinâmica de memória e Listas Encadeadas	Aula Teórica – Continuação de alocação dinâmica de memória. Introdução a listas encadeadas.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
13	17/09/2021	90	0	Listas encadeadas, pilhas e filas	Aula Teórica – Continuação de listas encadeadas com exemplos. Introdução a pilhas e filas.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
14	20/09/2021	0	60	Acompanhamento de Projeto	Acompanhamento individual de projeto e esclarecimento de dúvidas	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
15	21/09/2021	90	0	Recursão	Aula Teórica – Explorar o conceito de funções recursivas com exemplos.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
16	24/09/2021	90	0	Recursão e Revisão	Aula Teórica – Continuação dos conceitos de recursão. Revisão geral da unidade com exercícios.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
17	27/09/2021	0	60	Acompanhamento de Projeto	Acompanhamento individual de projeto e esclarecimento de dúvidas	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
18	28/09/2021	90	0	Revisão	Aula Prática – aula de revisão com exercícios	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)

					sobre todos os tópicos visto ao longo do semestre		
19	01/10/2021	90	0	PRIMEIRA AVALIACAO	Prova Pratica com uma lista de projetos na linguagem de programação C	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
20	04/10/2021	0	60	Acompanhamento de Projeto	Acompanhamento individual de projeto e esclarecimento de dúvidas	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
21	05/10/2021	90	0	Aula de Feedback e Conceitos Finais de C	Aula Teórica – Aula de feedback sobre a primeira unidade Conceitos finais de C – Pre-processadores, bibliotecas, leitura e escrita de arquivos.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
22	08/10/2021	90	0	Conceitos Finais de C e Introdução a Haskell	Aula Teórica – Como um programa em C é estruturado. Introdução a Haskell.e compilador Glasgow	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
23	11/10/2021	0	0	FERIADO	-----	-----	-----
24	12/10/2021	0	0	FERIADO	-----	-----	-----
25	15/10/2021	90	0	Introdução a Haskell e Listas	Aula Teórica – Continuação da Introdução de Haskell. Explicar como implementar listas em Haskell.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
26	18/10/2021	0	60	Acompanhamento de Projeto	Acompanhamento individual de projeto e esclarecimento de dúvidas	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
27	19/10/2021	90	0	Listas e Tuplas	Aula Teórica – Continuação sobre listas com exemplos. Introdução a tuplas.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
28	22/10/2021	90	0	Tuplas e Tipos Algébricos	Aula Teórica – Continuação sobre tuplas com exemplos. Descrição de tipos algébricos.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
29	25/10/2021	0	60	Acompanhamento de Projeto	Acompanhamento individual de projeto e esclarecimento de dúvidas	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
30	26/10/2021	90	0	Tipos Algébricos e Polimorfismo	Aula Teórica – continuação sobre Tipos Algébricos. Introdução a polimorfismo.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
31	29/10/2021	90	0	Polimorfismo e Type Class	Aula Teórica – Continuação de polimorfismo com exemplos. Introdução a Type Class	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
32	01/11/2021	0	0	FERIADO	-----	-----	-----
33	02/11/2021	0	0	FERIADO	-----	-----	-----
34	05/11/2021	90	0	Type Class e Funções de Alta Ordem	Aula Teórica – Continuação de Type Class com exemplos. Introdução a funções de alta Ordem	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)

35	08/11/2021	0	60	Acompanhamento de Projeto	Acompanhamento individual de projeto e esclarecimento de dúvidas	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
36	09/11/2021	90	0	Funções de Alta Ordem e Revisão	Aula Teórica – Continuação de funções de alta ordem. Revisão do assunto visto na unidade com exemplos.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
37	12/11/2021	0	60	Revisão	Aula Prática – Resolução de exercícios cobrindo todo o assunto visto na unidade.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
38	15/11/2021	0	0	FERIADO	-----	-----	-----
39	16/11/2021	0	60	Revisão	Aula Prática – Resolução de exercícios cobrindo todo o assunto visto na unidade.	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
40	19/11/2021	90	0	SEGUNDA AVALIACAO	Prova pratica com projetos de programação em Haskell	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
41	23/11/2021	90	0	Enceramento do Semestre	Aula final sobre a disciplina, com troca de ideias e feedbacks	Computador, Slides, Conexão Wifi, Zoom Meeting,	Sala de Aula Virtual (Zoom)
42	10/12/2021	0	0	SEGUNDA CHAMADA	-----	-----	-----
43	15/12/2021	0	0	AVALIACAO FINAL	-----	-----	-----
Total		2700	900				

Carga Horária	Minutos	Horas
Teórica	2700	45
Prática	900	15
Total	3600	60

Carga Horária	Minutos	Horas
Presencial	3600	60
Online	0	0
Total	3600	60

METODOLOGIA DE ENSINO APRENDIZAGEM

Metodologia de ensino baseada em aulas expositivas dialogadas, debates, estudos de caso, dinâmicas e apresentações de trabalhos por parte dos alunos, valorizando a dimensão crítica, participativa e construtiva com especial atenção à capacidade de expressão tanto oral quanto escrita. Para suporte às atividades do professor e alunos de projetos será utilizado um Ambiente Virtual de Aprendizado e um conjunto de ferramentas e recursos tecnológicas para comunicação com os alunos tais como: Slack, Email, Zoom, Google Meets, Projetor, Computador, Materiais em PDF e Videoaulas.

AVALIAÇÃO DOS PROCESSOS DE ENSINO-APRENDIZAGEM

O processo avaliativo de Programação Imperativa e Funcional dar-se-á nas diversas etapas do processo de aprendizagem de forma contínua e vinculada aos objetivos da disciplina. Esse processo será composto de listas de exercícios semanais e prova escrita no final de cada unidade.

As listas de exercícios serão lançadas semanalmente e consistem em uma ou duas questões práticas de nível básico a intermediário cobrindo o assunto visto naquela semana. Já as avaliações consistem em 5 ou mais questões de nível intermediário a avançado de todo o assunto visto ao longo da unidade. A soma das notas das listas corresponde a 50% da nota da unidade, enquanto a avaliação representa os demais 50%.

Além disto, no final do semestre, será considerada o nível de participação das atividades em sala de aula e uma avaliação final para os alunos que assim necessitarem.

A nota de cada unidade é descrita, portanto, da seguinte forma:

A nota do 1º Módulo será composta da seguinte maneira:

N1 = Prova (50%) + listas de exercícios em C (50%).

A nota do 2º Módulo será composta da seguinte maneira:

N2 = Prova (50%) + listas de exercícios em Haskell (50%).

BIBLIOGRAFIA BÁSICA POR UNIDADE CURRICULAR

BÁSICA

SCHILD, H., C Completo e Total. Pearson, 1997.

DEITEL, H. E DEITEL, P., C: Como Programar. Pearson, 2015

LIPOVACA, M., Learn You a Haskell for Great Good! No Starch Press, 2011.

BIBLIOGRAFIA COMPLEMENTAR

PETER VAN ROY, Programming Paradigms for Dummies: What Every Programmer Should Know,

JAMSA, K., Programando em C/C++ "A bíblia". Pearson, 1999.

BIRD, R., Thinking Functionally with Haskell. Cambridge University Press, 2014.

KERNIGHAN, B.W., C: a Linguagem de Programação. Campus, 1990.

Thompson, S., Haskell: The Craft of Functional Programming. Addison-Wesley Professional, 2011.

PERIÓDICOS ESPECIALIZADOS

ARTIGO

ARTIGO

ARTIGO

ARTIGO

JUSTIFICATIVA PARA ADOÇÃO DA BIBLIOGRAFIA

Os livros de Schildt e Deitel apresentam de forma clara, e sempre acompanhado de exemplos, tanto os elementos mais básicos bem como os mais avançados necessários para o aprendizado da linguagem C. De forma semelhante, o material construído por Lipovaca traz, em uma estrutura bastante didática, os conceitos e ferramentas presentes no Haskell, permitindo uma evolução gradual do aprendizado.