



UNIVERSIDADE FEDERAL DE OURO PRETO  
DEPARTAMENTO DE COMPUTAÇÃO



PLANO DE ENSINO

<b>Nome do Componente Curricular em português:</b> Programação Funcional		<b>Código:</b> BCC222
<b>Nome do Componente Curricular em inglês:</b> Functional Programming		
<b>Nome e sigla do departamento:</b> Departamento de Computação (DECOM)		<b>Unidade acadêmica:</b> ICEB
<b>Nome do docente:</b> Rodrigo Geraldo Ribeiro		
<b>Carga horária semestral:</b> 60 horas	<b>Carga horária semanal teórica:</b> 4 horas/aula	<b>Carga horária semanal prática:</b> 4 horas/aula
<b>Data de aprovação na assembleia departamental:</b> ___ / ___ / _____		
<b>Ementa:</b> Características dos principais paradigmas de programação; princípios do paradigma de programação funcional; principais características de linguagens de programação funcional: recursão, abstração funcional, funções de ordem superior, tipos de dados algébricos, polimorfismo, inferência de tipos, avaliação estrita e avaliação lazy, sobrecarga; estudo de uma linguagem funcional moderna e desenvolvimento de programas nesta linguagem, enfocando aspectos de correção, modularidade e reuso de código.		
<b>Conteúdo Programático:</b> <ul style="list-style-type: none"><li>• Introdução</li><li>• Paradigmas de programação</li><li>• Primeiros passos em haskell</li><li>• Definindo funções</li><li>• Tipos de dados</li><li>• Expressão condicional</li><li>• Funções recursivas</li><li>• Tuplas, listas e polimorfismo paramétrico</li><li>• Casamento de padrão</li><li>• Programas interativos</li><li>• Ações de e/s recursivas</li><li>• Números aleatórios e argumentos da linha de comando</li><li>• Arquivos</li><li>• Expressão lambda</li><li>• Funções de ordem superior</li><li>• Tipos algébricos</li><li>• Classes de tipos</li><li>• Mônadas</li></ul>		

- Avaliação lazy
- Prova de propriedades de programas

#### Objetivos:

Ao final do curso espera-se que os alunos possuam os seguintes conhecimentos e habilidades:

- Conhecimento das características fundamentais de linguagens funcionais modernas e noções básicas sobre o modelo de execução de programas nessas linguagens;
- Noções básicas sobre sistemas de tipos e inferência de tipos;
- Capacidade de distinção entre polimorfismo paramétrico, de sobrecarga e de inclusão;
- Entendimento dos diferentes mecanismos de avaliação em linguagens de programação;
- Capacidade para comparar características de linguagens de diferentes paradigmas;
- Habilidade para programar em uma linguagem funcional moderna;
- Entendimento sobre a implementação da noção de estado em linguagens funcionais;
- Noções de derivação de programas a partir da especificação e sobre prova de correção de programas em linguagens funcionais.

#### Metodologia:

Vídeos sobre o conteúdo teórico / prático da disciplina. Semanalmente serão fornecidos exercícios para suprir a carga horária de atividades práticas da disciplina.

#### Atividades avaliativas:

3 avaliações no valor de 10,0 pontos. A nota final é a média aritmética simples das notas obtidas nas 3 avaliações. As avaliações serão aplicadas de forma assíncrona. O exame especial será aplicado de forma assíncrona como um trabalho prático.

#### Cronograma:

Data prevista	Conteúdo
24/08/2020	Apresentação da disciplina: critérios de avaliação e ementa.
25/08/2020	Paradigmas de linguagens de programação.
26/08/2020	Introdução a linguagem Haskell: sintaxe, tipos, listas, tuplas.
27/08/2020	Declaração de funções e list comprehensions.
31/08/2020	Casamento de padrão e funções recursivas.
01/09/2020	Funções sobre listas.
02/09/2020	Polimorfismo paramétrico e funções de ordem superior.
03/09/2020	Funções de ordem superior e aplicação parcial.
07/09/2020	Funções de ordem superior e Funções anônimas.
08/09/2020	Estudo de caso: implementando a cifra de César.
09/09/2020	Revisão para avaliação 1.
10/09/2020	Avaliação 1
14/09/2020	Correção da avaliação 1. Tipos de dados algébricos.
15/09/2020	Definindo funções sobre tipos de dados algébricos.
16/09/2020	Estudo de caso: funções de ordem superior para árvores binárias.
17/09/2020	Polimorfismo de sobrecarga: Classes de tipos em Haskell.
21/09/2020	Definindo classes de tipos e instâncias.
22/09/2020	Classes de tipos da biblioteca padrão.
23/09/2020	Funtores aplicativos e monóides.
24/09/2020	Estudo de caso: códigos de Huffmann
28/09/2020	Revisão para avaliação 2
29/09/2020	Avaliação 2

30/09/2020	Correção da avaliação 2. Introdução às mônadas
01/10/2020	Mônadas e I/O em Haskell.
05/10/2020	Mônada de estado.
06/10/2020	Mônadas de somente leitura e somente escrita.
07/10/2020	Combinadores de parsing.
08/10/2020	Combinadores de parsing.
12/10/2020	Teste baseado em propriedades: QuickCheck
13/10/2020	Teste baseado em propriedades: QuickCheck
14/10/2020	Revisão para avaliação 3
15/10/2020	Avaliação 3
16/10/2020	Entrega de resultados parciais
19/10/2020	Exame especial.

#### **Bibliografia Básica:**

- Lipovaca, Miran - Learn you a Haskell for the great good! No Starch Press. Disponível gratuitamente on-line: <http://learnyouahaskell.com>
- O'Sullivan, Bryan; Stewart, Don ; Goerzen, John - Real World Haskell. Disponível gratuitamente on-line: <http://book.realworldhaskell.org>
- Jhala, Ranjit; Seidel, Eric ; Vazou, Niki - Programming with Refinement Types: An introduction to Liquid Haskell. Disponível gratuitamente on-line: <http://ucsd-progsys.github.io/liquidhaskell-tutorial/>

#### **Bibliografia Complementar:**

- SNOYMAN, Michael. Developing Web Applications with Haskell and Yesod. Disponível gratuitamente on-line: <https://www.yesodweb.com/book>
- MARLOW, Simon. Parallel and Concurrent Programming in Haskell. Disponível gratuitamente on-line: <https://simonmar.github.io/pages/pcph.html>
- WADLER, Philip; KOKKE, Wen; SIEK, Jeremy. Programming Languages Foundations in Agda. Disponível gratuitamente on-line: <https://plfa.github.io>