



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE SANTA MARIA

## PLANO DE ENSINO

Identificação da disciplina e dos dados da oferta
Código e nome da disciplina: ELC117 Paradigmas de Programação
Curso: Ciência da Computação
Turma:
Docente responsável: Juliana Kaizer Vizzotto
Ano/período: 2020/1
Objetivos da disciplina: Analisar os diferentes paradigmas de programação e selecionar o que melhor se adapte a solução do problema proposto.
Carga horária: 60h
Conteúdo programático: UNIDADE 1 - PROGRAMAÇÃO PROCEDIMENTAL 1.1 - Conceitos. 1.2 - Exemplos.  UNIDADE 2 - PROGRAMAÇÃO ORIENTADA A OBJETOS 2.1 - Conceitos da programação orientada a objetos. 2.2 - Exemplos de linguagens orientadas a objetos.  UNIDADE 3 - PROGRAMAÇÃO CONCORRENTE 3.1 - Processos concorrentes. 3.2 - Seções críticas. 3.3 - Exclusão mútua. 3.4 - Comunicação entre processos.  UNIDADE 4 - PROGRAMAÇÃO LÓGICA 4.1 - Introdução ao cálculo de predicados. 4.2 - Visão geral da programação lógica. 4.3 - Exemplos: Prolog. 4.4 - Aplicações da programação lógica.  UNIDADE 5 - PROGRAMAÇÃO FUNCIONAL 5.1 - Fundamentos de linguagens de programação funcionais. 5.2 - Exemplo de uma linguagem de programação funcional. 5.3 - Aplicações da programação funcional.
Bibliografia básica: SEBESTA, R. <b>Conceitos de linguagens de programação</b> . 9. ed. Porto Alegre: Bookman, 2011.
Bibliografia complementar: FINKEL, R. A. <b>Advanced programming language design</b> . Reading: Addison-Wesley, 1996.



FISCHER, A. E. & GRODZINSKY, F. S. **Anatomy of programming languages.** Englewood Cliffs: Prentice-Hall, 1993.

JAZZAYERI, M. & GHEZZI, C. **Conceitos de linguagens de programação.** 3. ed. Rio de Janeiro: Campus, 1998.

### Descrição do plano

#### Metodologia:

Durante o semestre teremos aulas expositivas em sala de aula, atividades práticas de programação no laboratório, apresentação de seminários com debate e leitura de artigos científicos.

#### Cronograma de atividades<sup>1</sup>:

Seg 09/03/2020: Introdução e motivação. Paradigmas de linguagens de programação, conceitos básicos, compiladores vs interpretadores, sintaxe e semântica, sistemas de tipos.

Qua 11/03/2020: Fundamentos de linguagens de programação funcional. Funções na linguagem Haskell.

Seg 16/03/2020: Programação funcional: funções de alta ordem, funções anônimas, pattern matching, avaliação lazy.

Qua 18/03/2020: Programação funcional: funções de alta ordem e funções anônimas (prática).

Seg 23/03/2020: Programação funcional: listas e list comprehension.

Quar 25/03/2020: Programação funcional: polimorfismo, classes de tipos .

Seg 30/03/2020: Programação funcional: recursividade.

Qua 01/04/2020: Programação funcional: let ...in, where e IO.

Seg 06/04/2020: Programação lógica.

Qua 08/04/2020: Programação lógica.

Seg 13/04/2020: Programação lógica

<sup>1</sup>Destacar atividades que serão desenvolvidas em situação de docência orientada, quando houver; as atividades a distância, quando previstas no PPC e aprovadas nos respectivos colegiados de curso; bem como os eventuais afastamentos do professor.



Qua 15/04:ERAD 2020.

Seg 20/04: Dia não letivo.

Qua 22/04: Programação em lógica.

Seg 27/04: Avaliações do primeiro bimestre.

Qua 29/04: Avaliações do primeiro bimestre.

Seg 04/05: Programação orientada a objetos.

Qua 06/05: Programação orientada a objetos.

Seg 11/05: Programação orientada a objetos.

Qua 13/05: Programação orientada a objetos.

Seg 18/05: Programação orientada a objetos.

Qua 20/05: Programação orientada a objetos.

Seg 25/05: Programação orientada a objetos.

Qua 27/05: Programação orientada a objetos.

Seg 01/06: Programação orientada a objetos.

Qua 03/06: Programação orientada a objetos.

Seg 08/06: Programação orientada a objetos.

Qua 10/06: Programação orientada a objetos.

Seg 15/06: Programação Concorrente.

Qua 17/06: Programação Concorrente.



Seg 22/06: Programação Concorrente.

Qua 24/06: Finalização dos trabalhos.

Seg 29/06: Finalização dos trabalhos.

Qua 01/07: Avaliações segundo bimestre.

Seg 13/07: Avaliação Final (Exame).

**Atividades práticas:**

A disciplina terá diversas atividades práticas de programação, que serão executadas no laboratório ou em casa.

**Critérios de avaliação:**

O método de avaliação da disciplina se dará através da realização de duas provas, com peso 4 no primeiro bimestre e com peso 3 no segundo bimestre, e através da realização de trabalhos práticos de implementação, bem como apresentação de seminários.

A nota final da disciplina será calculada em duas etapas:

$$B1 \text{ (bimestre 1)} = P1(\text{prova 1}) * 0.4 + T1s \text{ (trabalhos do bimestre 1)} * 0.6$$

$$B2 \text{ (bimestre 2)} = P2(\text{prova 2}) * 0.3 + T2s \text{ (trabalhos do bimestre 2)} * 0.7$$

$$\text{Final} = (B1 + B2) / 2$$

No primeiro bimestre serão dois trabalhos práticos, Haskell 20% e Prolog 20%, e mais um seminário sobre algum assunto a definir com a professora com peso de 20%.

No segundo bimestre será um trabalho de jogo de aventura no paradigma orientado a objeto com duas etapas de avaliação, cada etapa com peso 35%.

As datas das provas se encontram no cronograma acima.

**Informações complementares:**

**Bibliografia complementar:**

Bruce A. Tate: Seven languages in seven weeks. Pragmatic Bookshelf, 2010.



**MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE SANTA MARIA**

---

Nome

Juliana Kaizer Vizzotto

---

Nome e carimbo

Coordenador(a) do Curso