

Instituto de Informática
Departamento de Informática Teórica

Dados de identificação

Disciplina: TÓPICOS ESPECIAIS EM COMPUTAÇÃO V

Período Letivo: 2014/1

Período de Início de Validade: 2014/1

Professor Responsável pelo Plano de Ensino: RODRIGO MACHADO

Sigla: INF05504

Créditos: 2

Carga Horária: 30

Súmula

Assuntos relacionados a inovações teóricas decorrentes de pesquisas recentes, aplicações específicas, interessando a um grupo restrito ou tendo caráter de temporariedade, aspectos abordados superficialmente em disciplinas regulares.

Currículos

| Currículos | Etapla Aconselhada | Natureza |
|--------------------------------------|--------------------|----------|
| BIOTECNOLOGIA MOLECULAR | | Eletiva |
| BIOINFORMÁTICA | | Eletiva |
| BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO | | Eletiva |

Objetivos

Capacitar os alunos a utilizar linguagens de programação funcionais para o desenvolvimento de sistemas de complexidade média/alta. Apresentar técnicas de programação voltadas a linguagens funcionais.

Conteúdo Programático

| |
|---|
| Semana: 1 a 2 |
| Título: Revisão de programação funcional |
| Conteúdo: Linguagens funcionais puras e impuras. Avaliação estrita e preguiçosa. Valores numéricos, listas, tuplas. Recursão simples. Recursão múltipla. Funções de alta ordem. Exemplos avançados do uso de recursão. |
| Semana: 3 a 5 |
| Título: Introdução à linguagem Haskell e ao sistema de tipos Hindley/Milner |
| Conteúdo: Introdução à linguagem Haskell. Sintaxe. Tipos básicos. Funções anônimas. Sintaxe de compreensão de lista (list comprehension). Tipos de dados algébricos. Casamento de padrões (pattern matching). Sinônimos de tipo. Tipos de dados parametrizados. Introdução a sistemas de tipo polimórficos. Sistema de tipos Hindley/Milner. |
| Semana: 6 a 7 |
| Título: Construtores de tipos e typeclasses |
| Conteúdo: Construtores de tipos comuns em Haskell: Maybe, Either, tuplas, listas, funções. Typeclasses. Classes básicas: Eq, Ord, Num, Read, Show |
| Semana: 8 a 10 |
| Título: Mônadas e aplicações |
| Conteúdo: Mônadas: definição e exemplos básicos: Id, Maybe, List, State. Entrada e saída de dados. Números randômicos. Biblioteca Parsec (combinadores de parsers). |
| Semana: 11 a 12 |
| Título: Dialetos imperativos e orientados a objeto em linguagens funcionais. |
| Conteúdo: Programação no estilo imperativo em Haskell. Arrays. Programação no estilo orientado a objetos em Haskell. Uso de herança via typeclasses. |
| Semana: 13 |
| Título: Otimização. |
| Conteúdo: Técnicas de otimização de código em linguagens com avaliação preguiçosa. |
| Semana: 14 |
| Título: Bibliotecas gráficas. |
| Conteúdo: APIs gráficas em Haskell para construção de GUIs e renderização 2d (Cairo) e 3d (OpenGL). |
| Semana: 15 |

| | |
|--|---|
| Título: | Apresentação de projetos. |
| Conteúdo: | Apresentação de projetos de programação desenvolvidos durante o curso. |
| Metodologia | Aulas expositivas. Listas de exercícios. Desenvolvimento de um projeto de programação ao longo do semestre. |
| Carga Horária | Teórica: 30 Prática: 0 |
| Experiências de Aprendizagem | Implementações de pequenos programas (listas de exercícios extra-classe). Desenvolvimento de um projeto de programação a ser apresentado ao final do semestre, utilizando as técnicas vistas em aula. |
| Critérios de avaliação | Os alunos serão avaliados pelo projeto desenvolvido ao longo do semestre. Haverá duas etapas: 1: validação do experimento de programação, onde o escopo e a funcionalidade a ser implementada será definida (metade do semestre) 2 : apresentação do sistema implementado para o professor e a turma, onde o sistema em si e as técnicas utilizadas serão apresentados (final do semestre) A esse projeto será dado um conceito (A – excelente, B – bom, C – minimamente satisfatório ou D – insatisfatório). O projeto poderá ser realizado individualmente ou em duplas. Se o aluno não obtiver frequência mínima de 75% da carga horária, terá conceito FF (falta de frequência). |
| Atividades de Recuperação Previstas | Para projetos que tenham recebido conceito D, C ou B, há a possibilidade de acréscimo de um nível do conceito (D → C; C → B ou B → A) condicionado à entrega posterior de correções ou extensões do código apresentado. Nesse caso, não será necessário nova apresentação. |
| Bibliografia | |
| Básica Essencial | <i>Sem bibliografias acrescentadas.</i> |
| Básica | Bryan O'Sullivan, Don Stewart, and John Goerzen. Real World Haskell. O'Reilly, ISBN 978-0-596-51498-3. Disponível em: http://bookrealworldhaskell.org/ Miran Lipovaca. Learn you a Haskell for great good! A beginner's guide.. No starch press, ISBN 978-1-59327-283-8. Disponível em: http://learnyouahaskell.com Pierce, Benjamin C.. Types and programming languages. MIT Press, ISBN 0262162091. Disponível em: http://ieeexploreieee.org/xpl/bkabstractplusjsp?bkn=6267321 Simon Marlow. Haskell 2010 Language Report. Disponível em: http://www.haskell.org/onlinereport/haskell2010/ |
| Complementar | <i>Sem bibliografias acrescentadas</i> |
| Outras Referências | <i>Não existem outras referências para este plano de ensino.</i> |
| Observações | Nome da disciplina: Programação Funcional Avançada. - As 30 horas previstas para atividades teóricas e práticas indicadas neste Plano de Ensino incluem 15 encontros de 100 minutos de |

duração (2 períodos de 50 minutos por encontro, 1 encontro por semana, durante 15 semanas), num total de 1.500 minutos, e mais 5 horas (300 minutos) de atividades autônomas, realizadas sem contato direto com o professor, correspondentes a exercícios e trabalhos extraclasse;

- O Professor poderá se valer de aulas presenciais ou à distância (recursos de EAD);
- A disciplina poderá contar com o apoio de Professores Assistentes (alunos de Pós-Graduação) em Atividade Didática.