

UFRN/DIMAp

Anexo I

MODELO DE PLANO DE CURSO PARA COMPONENTE CURRICULAR OFERECIDO PELO DIMAp, EXCLUSIVO PARA A RETOMADA DO PERÍODO LETIVO 2020.1 (COMO 2020.6) NO FORMATO REMOTO

Dados do Componente	
Unidade responsável	Departamento de Informática e Matemática Aplicada
Código	DIM0437
Nome	Linguagens de Programação: Conceitos e Paradigmas
Carga horária teórica	60h
Carga horária prática	0h

Dados do docente*	
Nome: Umberto Souza da Costa	SLAPE: 1495704
Cargo: Professor de Ensino Superior	
Unidade de exercício: Departamento de Informática e Matemática Aplicada	
email: umberto@dimap.ufrn.br	

*No caso de componente curricular a ser ofertado por mais de um docente, o quadro anterior deve ser replicado.

Conteúdo	Em caso de componente curricular já cadastrado, copie a ementa do SIGAA (na aba Ensino > Consulta > Componentes curriculares
	Sintaxe e semântica. Tipos e valores. Variáveis e amarração (binding). Sub-rotinas e Co-rotinas. Passagem de Parâmetros. Recursividade, paralelismo, sincronização. Paradigmas de programação: imperativo, funcional, lógico, orientado a objetos. Estudo comparativo de linguagens.

Metodologia	Descrição de como a disciplina será desenvolvida, especificando as técnicas de ensino a serem utilizadas. Disciplinas com carga horária prática devem atentar ao disposto no Art. 2º "Os componentes curriculares de natureza prática ou a parte prática de componentes curriculares poderão ser adaptados ao formato remoto, desde que seja elaborado plano de curso específico, para o período letivo 2020.1, aprovado pelo colegiado de curso e apensado ao Projeto Pedagógico de Curso." Sendo assim, quando for o caso, deve ficar claro como a carga horária prática será adaptada ao formato remoto.
	A disciplina adotará a Aprendizagem Baseada em Problemas (PBL) para explorar tantos os aspectos teóricos quanto seu projeto de implementação. Nesta abordagem, os alunos serão organizados em pequenos grupos, estes grupos serão acompanhados pelo docente e serão responsáveis por resolver problemas utilizados como estímulo à aquisição de conhecimentos e compreensão de conceitos. Desta forma, a metodologia da disciplina aproxima-se àquela do mercado de trabalho. O processo de aprendizagem em PBL tem como princípios: (i) ativar o conhecimento prévio do estudante; (ii) identificar as necessidades de aprendizagem; (iii) desenvolver no aluno a capacidade de criticar antigos e novos conhecimentos e de construção de uma nova síntese que possa ser aplicada a outras situações; (iv) não há exposição formal prévia de informação. Para cada problema, os seguintes passos deverão ser executados: (1) esclarecimento do problema/situação apresentada; (2) exploração e análise integrada dos dados do problema; (3) identificação do conhecimento atual relevante para o problema – brainstorm; (4) desenvolvimento de questões de aprendizagem; (5) busca de novos

UFRN/DIMAp

	<p>conhecimentos; (6) síntese dos conhecimentos prévios e novos – como se pode solucionar o problema agora? (7) resolução do problema.</p> <p>Os problemas a serem abordados pelo grupo, a indicação de referências bibliográficas e a submissão de soluções serão realizadas da maneira costumeira, ou seja, por meio da turma virtual do SIGAA. Por outro lado, utilizaremos o aplicativo Discord, para viabilizar que os grupos se reúnam remotamente, com o acompanhamento do docente. O Discord permite a criação de salas virtuais, uma para cada grupo de alunos, dentro de um mesmo canal, a ser associado à turma. A comunicação pelo Discord poderá ser feita via mensagens de texto, áudio e compartilhamento de telas, podendo o docente alternar entre salas virtuais para acompanhar as discussões dos grupos. Alternativamente, podemos utilizar o Google Meet nas aulas.</p>
--	--

Procedimentos de avaliação da aprendizagem	Descrição dos instrumentos e critérios a serem utilizados para a verificação da aprendizagem.
	<ul style="list-style-type: none"> Soluções aos problemas propostos, submetidas via SIGAA: 30% Implementação do interpretador: 70% OBS.: as porcentagens acima referem-se à nota global do período letivo.

Cronograma e critérios para validação da assiduidade dos discentes	Cronograma detalhado das atividades e dos critérios de validação da assiduidade dos discentes.
	<p>(Art. 3º §4º "A frequência e a participação dos discentes serão verificadas de acordo com o acompanhamento das atividades propostas, conforme plano de curso.")</p> <p>A assiduidade dos alunos será aferida de acordo com a entrega das soluções aos problemas (Problemas 1 a 7) e pela entrega da implementação do interpretador. As datas de entrega desses documentos/produtos estão definidas no cronograma de aulas síncronas (ver quadro a seguir).</p>

Datas e horários das atividades síncronas	Em caso de atividade síncrona, indicar o momento em que os encontros acontecerão, de acordo com o cronograma. Essa informação é importante para que os estudantes possam avaliar sua participação na turma.
	Vale salientar que qualquer atividade síncrona deve acontecer no horário previamente cadastrado para a turma, conforme indicado no Art. 3º, §2º "Para as atividades de interação online síncronas com os discentes, previstas nos planos de curso, os docentes deverão respeitar os dias e horários registrados para a turma no SIGAA".
	<p>Abaixo, segue o cronograma das aulas síncronas da turma:</p> <ul style="list-style-type: none"> 24/08, M56: Apresentação da disciplina, Reorganização de grupos 26/08, M56: Revisão do Problema 1 (concepção da linguagem de programação) e do Paradigma Funcional. 31/08 a 09/09, M56: Problema 2 (definição da linguagem de programação) 14/09, M56: Protótipo do analisador léxico da linguagem proposta 16/09 a 23/09, M56: Problema 3 (nomes, variáveis, valores e tipos) 28/09, M56: Representação dos tipos da linguagem proposta em Haskell 30/09 a 05/10, M56: Problema 4 (vinculações, escopos e sistema de tipos) 07/10, M56: Protótipo do analisador sintático da linguagem proposta 14/10 a 21/10, M56: Problema 5 (blocos, comandos, expressões e estruturas de controle) 26/10, M56: Protótipo da tabela de símbolos e funções relacionadas 04/11 a 11/11, M56: Problema 6 (subprogramas) 16/11, M56: Protótipo do avaliador de expressões e verificador de tipos 18/11 a 23/11, M56: Problema 7 (abstrações de larga escala) 25/11 a 02/12, M56: Integração dos interpretadores Haskell 07/12 a 09/12, M56: Avaliação dos Interpretadores 14/12, M56: Avaliação de Reposição <p>As soluções de cada problema deverão ser submetidas (via SIGAA) até a</p>

UFRN/DIMAp

véspera de início da próxima atividade.

Compensação de Conteúdo	Descrição de como o conteúdo já ministrado nas semanas antes da suspensão das aulas será compensado, quando for o caso. (Art. 3º §5º "Para turmas já iniciadas, deverá ser realizada a compensação de conteúdo.")
	Antes da suspensão das atividades em 2020.1, foram ministradas 7 (sete) aulas. Estas 7 (sete) aulas corresponderam à apresentação da disciplina, resolução do Problema 1 e introdução aos paradigmas de programação Lógico e Funcional. Durante a primeira semana de 2020.6, realizaremos a revisão da maior parte destes conteúdos. Ainda nesta primeira semana, reorganizaremos os grupos para acomodar eventuais desistências ou ingressos de novos alunos na turma. Para eventuais novos alunos na turma, novos prazos serão dados para a submissão das soluções do Problema 1.

Detalhamento dos recursos didáticos a serem utilizados	Descrição dos recursos didáticos a serem utilizados pelo docente para a realização das atividades.
	Em sua maioria, as aulas deverão ser guiadas pelos problemas elaborados e disponibilizados antecipadamente pelo docente como documentos PDF, via SIGAA. Cada problema deverá incluir referências relevantes, a serem consultadas pelos alunos antecipadamente para posterior discussão nos encontros síncronos (via os aplicativos Discord ou Google Meet). Eventualmente, slides poderão ser disponibilizados no SIGAA para esclarecer conceitos mais complexos. Conceitos e questões de implementação poderão ser esclarecidos também por meio do compartilhamento de tela entre docente e estudantes.

Recursos necessários para o acompanhamento da turma pelo discente	Descrição dos recursos necessários para que o discente possa acompanhar as atividades da turma de forma adequada, por exemplo, plataformas de hardware e/ou software, requisitos computacionais e demais recursos necessários para realizar as atividades programadas.
	<ul style="list-style-type: none">• Acesso à internet nos horários das atividades síncronas;• Dispositivo (celular, tablet ou computador) disponível durante o horário das aulas síncronas, com capacidade de comunicação por áudio e, idealmente, também por vídeo;• Instalação do aplicativo Discord e Google Meet em seu dispositivo;• Computador com Ubuntu 16.04 LTS (ou superior) ou com Mac OS, incluindo a instalação da plataforma Haskell (https://www.haskell.org/) e das ferramentas Alex (https://www.haskell.org/alex/) e Parsec (https://hackage.haskell.org/package/parsec), para criação do interpretador para a linguagem definida pelo grupo.• Outros aplicativos gratuitos, caso conveniente.

Materiais e Referências	<p>Descrição dos materiais próprios ou de curadoria a serem utilizados para a realização das atividades, explicitando a forma de disponibilização para os discentes.</p> <p>Indicar referências a serem utilizadas para a realização das atividades, dando preferência a materiais gratuitos, online e acessíveis aos discentes.</p> <p>(Art. 5º Os materiais didáticos deverão ser disponibilizados pelos docentes durante todo o período, considerando as limitações das condições de isolamento social impostas pela pandemia da COVID-19.</p> <p>§2º Quando necessário, os materiais utilizados nas atividades e/ou a forma de comunicação devem ser adaptados, de forma a atender discentes com algum tipo de deficiência e/ou com necessidades educacionais específicas, considerando as orientações da Secretaria de Inclusão e Acessibilidade (SIA).)</p> <ul style="list-style-type: none"> • LEE, Kent D. Foundations of Programming Language. Second Edition, Springer. Disponível em: https://doi.org/10.1007/978-3-319-70790-7 • Florida Tech, CS: Programming Language Concepts (Spring 2020): Disponível em: https://cs.fit.edu/~ryan/cse4250/ • RANGEL, José Lucas. Notas de aula. Disponível em http://www.ic.uff.br/~cbraga/lf/rangel.zip. • WATT, David A. FINDLAY, William. Programming Language Design Concepts. John Wiley & Sons Ltd, 2004. • SEBESTA, Robert. Conceitos de Linguagens de Programação. Bookman: 2003. • VAREJÃO, Flávio. Linguagens de Programação. Elsevier: 2004. • MELO, Ana Cristina Vieira. SILVA, Flávio Soares Corrêa. Princípios de Linguagens de Programação. Edgard Blücher: 2003.
--------------------------------	---

Informações adicionais:	<p>Acrescente aqui informações relevantes sobre o seu Plano de Curso e o desenvolvimento das atividades da turma.</p> <p>Esta disciplina tem como principal objetivo abordar conceitos relevantes à concepção e implementação de linguagens de programação imperativas. Na disciplina, os grupos de alunos deverão estudar estes conceitos e aplica-los à concepção e implementação de uma linguagem definida pelo grupo.</p> <p>As atividades síncronas serão dedicadas ao acompanhamento dos grupos pelo docente, de forma a corrigir eventuais problemas e consolidar conhecimentos. Contudo, a Aprendizagem Baseada em Problemas (PBL) exigirá que os componentes de cada grupo se reúnam em horários além das aulas síncronas, tanto para estudo e discussão das referências recomendadas como para elaborar e implementar a linguagem proposta. Desta forma, é importante que o aluno reserve tempo para estudo além do previsto pelas atividades síncronas.</p>
--------------------------------	--

Natal/RN, 31 de julho de 2020.



Umberto Souza da Costa