



Universidade Federal de Goiás
Instituto de Informática
Bacharelado em Engenharia de Software
Integração 1

Componente Curricular

| Código Disc. | Nome | Carga Horária | |
|--------------|--------------|---------------|---------|
| | | Teórica | Prática |
| 51 | Integração 1 | 32 | 32 |

Turma A: 2020/1

Professor: Fábio Nogueira de Lucena

EMENTA DA DISCIPLINA

1. Rever, exemplificar o emprego da Engenharia de Software em abrangência e profundidade.
2. Integrar todo o conhecimento das disciplinas do curso de tal forma a permitir a compreensão, a relação entre elas, a importância, os produtos e atividades pertinentes a cada uma delas.

OBJETIVOS DA DISCIPLINA

Objetivo geral

Fornecer e elucidar uma visão coesa e integrada da Engenharia de Software de tal forma que o estudante fortaleça sua percepção de como fazer uso do conhecimento desta área em toda a sua extensão.

Objetivos específicos

- Revisitar conhecimento em Engenharia de Software.
- Ambientação com projeto real de “destaque”.

| PROGRAMA | | |
|---|---|---|
| <ul style="list-style-type: none"> • Visão geral do MPS.BR • Visão geral do SWEBOK (v3) • Requisitos de software. • Projeto de software. • Construção de software. • Testes de software. • Manutenção de software. • Gerenciamento de configuração. | | |
| LEGENDA DOS PROCEDIMENTOS DIDÁTICOS | | |
| Legenda | Descrição | Objetivo |
| AEX | Aula teórica | Transmitir conhecimento utilizando quadro ou slides. |
| AP | Aula prática | Proporcionar ao aluno a aplicação prática do conteúdo ministrado em aula teórica. |
| ED | Estudo dirigido | Desenvolver a capacidade analítica, capacidade de síntese, de avaliação crítica e de análise. |
| OTR | Outros | Transmitir conhecimento utilizando quadro ou slides. |
| RE | Aula teórica com resolução de exercícios | Desenvolver o raciocínio lógico, criatividade e capacidade de abstração e a capacidade de identificar, analisar e projetar soluções de problemas. |
| SE | Seminários | Desenvolver o raciocínio lógico, criatividade, capacidade de abstração, capacidade para identificar, analisar, projetar soluções de problemas, a capacidade de comunicação oral e a capacidade de trabalhar em grupo. |
| TG | Trabalho em grupo | Desenvolver a capacidade de comunicação oral e escrita. Capacidade de trabalhar em grupo. |
| FERRAMENTAS E RECURSOS PARA O ENSINO REMOTO | | |
| Fórum | Pode ser utilizado isolado ou associado a outras ferramentas em atividades dirigidas. Nele, os participantes podem expressar sua opinião. É uma ferramenta assíncrona. Sua utilização considera aspectos qualitativos e quantitativos. | |
| Diário | Permite aos participantes postarem suas reflexões acerca de um tema e o relato, de modo assíncrono, do seu processo de aprendizagem. Possibilita a interação apenas entre estudante e professor-tutor. | |
| Wiki | Ferramenta assíncrona de escrita colaborativa. Permite edição coletiva dos documentos e atualização dinâmica. É necessário estar articulada a outra ferramenta, como o fórum e o <i>chat</i> , para que os participantes possam organizar suas ideias e traçar suas metas. | |
| Chat | Ferramenta de comunicação síncrona, exigindo que os participantes da discussão estejam conectados simultaneamente, de modo síncrono, para que o processo de comunicação seja efetuado; bate-papo por meio de aplicativos de <i>instant message</i> (exemplo: Whats app, Telegram, Discord, etc.). | |
| Lista de discussão | Ferramentas de comunicação assíncronas. Caracteriza-se pelo recebimento e envio de mensagens por e-mail. | |
| Blog/ Videoblog | Páginas pessoais da Internet cujo mecanismo possibilita registrar e atualizar em ordem cronológica, opiniões, fatos, emoções, imagens, além de outros conteúdos que se queira disponibilizar; | |

| | |
|-------------------------------|--|
| Tarefa | Consiste na descrição ou no enunciado de uma atividade a ser desenvolvida pelo participante, de modo assíncrono, que pode ser enviada em formato digital ao servidor da plataforma e será verificada posteriormente pelo professor-tutor; |
| Aula remota | Aula (preleção, demonstração, orientação) mediada por ambiente virtual em que professor e estudantes participam de modo síncrono, contendo recursos de interatividade (áudio, vídeo e/ou <i>chat</i>). Exemplos: Google Meet, MS Teams, Zoom, BBB, Whereby, etc.) |
| Glossário | Ferramenta que permite criar e atualizar uma lista de definições como em um dicionário, de modo assíncrono. |
| Questionário | Ferramenta assíncrona de recepção de materiais, possibilitando ao estudante submeter arquivos digitais, para posterior avaliação e <i>feedback</i> do professor. |
| Podcast/ Videocast | Ato de difundir mensagens de áudio e/ou vídeo pela Internet (ou de modo listado) em servidores de vídeos (Youtube, SnapTube, etc.). Permite ao participante criar arquivos de áudio ou vídeo para publicação e/ou difusão de conhecimentos, episódios, demonstrações, etc. |
| Enquete online | Recurso de interação para obtenção de respostas acerca de um tema, pesquisa ou questão posta no modo síncrono. |

PROCEDIMENTOS METODOLÓGICOS

Os tópicos (programa) da disciplina serão abordados por meio da indicação de material correspondente e, para cada um deles, atividades serão definidas. As atividades podem incluir várias modalidades e a escolha delas irá variar conforme o tópico e os insumos coletados com o andamento da disciplina. **Convém observar que o ensino remoto reflete um momento excepcional que demanda uma abordagem também excepcional, detalhada doravante.**

Toda quarta-feira, das 19h às 20h ocorrerá o momento síncrono da disciplina. Este momento síncrono contará como 25% da carga horária da disciplina e, em particular, 25% das presenças. Ele ocorrerá por meio da sala indicada abaixo:

<https://meet.google.com/lookup/b53ap7ppm2>

OS OUTROS 75% DA DISCIPLINA SERÁ ASSÍNCRONO. A interação com o docente ocorrerá por meio do repositório Github criado especificamente para esta finalidade: <https://github.com/kyriosdata/integracao>. Em tempo, este também é o portal e o Ambiente Virtual de Aprendizagem a ser utilizado no presente curso. Neste portal também estará detalhado, para cada tópico, as atividades correspondentes, o prazo para execução e os artefatos a serem produzidos.

Adicionalmente aos tópicos será realizado um trabalho em grupo. Estudantes serão distribuídos em grupos e para cada grupo será atribuído um estudo de caso real (repositório no Github) sobre o qual o objetivo da disciplina será trabalhado. Cada grupo deverá indicar um software cujo repositório no Github tenha pelo menos 50k "estrelas".

Cabe ao docente aprovar o software (repositório) indicado pelo grupo. Caso não possua um projeto de preferência você pode consultar <https://gitstar-ranking.com/>.

Cada grupo fará uso de um estudo de caso (software) distinto dos demais. De forma resumida, a intenção é *analisar um projeto de sucesso à luz do conhecimento da Engenharia de Software*. Noutras palavras, o foco não está exatamente no conhecimento da Engenharia de Software nem tampouco em projetos de sucesso, mas em como tais projetos fazem uso da Engenharia de Software.

A intenção é ambientar-se acerca de "como se desenvolve software hoje em dia?"
Consulte o link <https://github.com/personal> para uma ambientação inicial.

CRITÉRIO DE AVALIAÇÃO

A avaliação de desempenho e também da frequência (presenças) do estudante na disciplina é determinada pelo registro dos artefatos definidos para cada uma das atividades previstas. O registro no Github dos artefatos correspondentes e, naturalmente, o conteúdo destes artefatos será empregado na definição do desempenho e da frequência do estudante na disciplina.

A avaliação é contínua. A avaliação produz um resultado COM APROVEITAMENTO ou SEM APROVEITAMENTO. A nota final na disciplina será definida com base no total de atividades para as quais o resultado foi COM APROVEITAMENTO. Por exemplo, se forem definidas 10 atividades e você teve aproveitamento em 6 ou mais delas, então será considerado aprovado. Caso contrário, será considerado reprovado por média. Formalmente, a nota final é a razão entre o total de atividades com aproveitamento e o total de atividades, multiplicada por 10.

Toda atividade tem um prazo bem-definido para a sua realização. Atividade realizada fora do prazo não será considerada, nem na perspectiva do desempenho nem tampouco da frequência. Mesmo que não tenha aproveitamento em uma atividade, conforme avaliação dos docentes, receberá a frequência correspondente.

Observações:

(1) Estará aprovado o(a) aluno(a) que atingir nota final (NF) igual ou superior a 6.0 (seis pontos) e frequência às atividades acadêmicas igual ou superior a 75% da carga horária da disciplina. Se tivermos 10 tópicos, então terá que ter aproveitamento de suas atividades em pelo menos 8 destes tópicos.

(2) Será atribuída a nota 0 (zero) a qualquer atividade avaliativa não realizada na data estipulada. Adicionalmente, a presença correspondente (frequência), não será computada, neste caso.

(3) Todas as atividades assíncronas são supervisionadas. Os docentes irão acompanhar o que foi feito ou não, por meio do Github.

(4) Os alunos que se envolverem em plágio (desvios de conduta, seja como facilitador ou como beneficiário) receberão nota 0 (zero) para a atividade correspondente. O caso será levado ao conhecimento da Coordenação do Curso, do Núcleo Docente Estruturante e do Conselho Diretor do Instituto de Informática para as providências cabíveis e legais. O pedido de segunda chamada deverá ser protocolado conforme condições estipuladas na Resolução CONSUNI específica (RGCG) em vigor.

(5) Este Plano está amparado pelas normativas e portarias emanadas dos órgãos governantes superiores, pelas resoluções, instruções normativas e diretrizes didático-pedagógicas da UFG e do INF, em vigor, que definem e regulam o funcionamento do ensino remoto excepcional.

BIBLIOGRAFIA

Bibliografia Básica

1. Guia Geral MPS de Software (MpsBR) ([aqui](#))
2. Software Engineering Body of Knowledge ([SWEBOK v3](#))

Bibliografia Complementar

1. *Sommerville, Ian. Engenharia de Software - 8a edição - Addison Wesley. 2007*
2. *Braude, Eric J. Software Engineering - An Object-Oriented Perspective. John Wiley & Sons, c2001*
3. *Pfleeger Shari. Software Engineering - Theory and Practice. Prentice Hall, 2001.*