

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO RIO GRANDE DO SUL
CAMPUS RESTINGA

DEISE PINHEIRO VICTORINO

Uma Análise sobre o Ensino Atual da Engenharia de Software
frente ao uso dos Métodos Ágeis no Desenvolvimento de Software

Porto Alegre

2021

DEISE PINHEIRO VICTORINO

Uma Análise sobre o Ensino Atual da Engenharia de Software
frente ao uso dos Métodos Ágeis no Desenvolvimento de Software

Trabalho de Conclusão de Curso apresentado,
junto ao curso de Análise e Desenvolvimento
de Sistemas do Instituto Federal de Educação,
Ciência e Tecnologia do Rio Grande do Sul,
como requisito parcial para a obtenção do grau
de Tecnólogo em Análise e Desenvolvimento
de Sistemas.

Orientadora: Prof^a. Dr^a. Eliana Beatriz Pereira

Porto Alegre

2021

DEISE PINHEIRO VICTORINO

Uma Análise sobre o Ensino Atual da Engenharia de Software
frente ao uso dos Métodos Ágeis no Desenvolvimento de Software

Trabalho de Conclusão de Curso apresentado
como requisito parcial para a obtenção do grau
de Tecnólogo em Análise e Desenvolvimento
de Sistemas.

Aprovada em ____ / ____ / ____

BANCA EXAMINADORA

Prof.^a. Dr.^a. Eliana Beatriz Pereira (Orientadora)
Instituto Federal do Rio Grande do Sul

Prof. Dr. Iuri Albandes Cunha Gomes (Banca)
Instituto Federal do Rio Grande do Sul

Prof. Dr. Gleison Samuel do Nascimento (Banca)
Instituto Federal do Rio Grande do Sul

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO SUL

Reitor: Júlio Xandro Heck

Pró-reitor de Ensino: Lucas Coradini

Diretor Geral: Prof. Rudinei Müller

Coordenador do CST em Análise e Desenvolvimento de Sistemas: Prof. Iuri
Albundes Cunha Gomes

Bibliotecária-chefe do Campus Restinga: Paula Porto Pedone

Dedico esse trabalho aos meus familiares e amigos por sempre me apoiarem em todos os momentos.

AGRADECIMENTOS

Agradeço à professora orientadora Eliana Beatriz Pereira por me orientar neste trabalho mostrando o melhor caminho a ser seguido, bem como mostrar o que poderia ser melhorado. Gostaria de agradecer especialmente meus pais por sempre acreditarem e cuidarem de mim durante toda minha vida, me incentivando a estudar e explorar mais o mundo do conhecimento, possibilitando essa jornada acadêmica e também ao meu esposo pelo incentivo e paciência nos momentos em que precisei.

RESUMO

Após o Manifesto Ágil, a utilização dos Métodos Ágeis tornou-se uma realidade no desenvolvimento de software fazendo com que a academia adequasse seu currículo para atender as necessidades da indústria. Este Trabalho de Conclusão de Curso tem como objetivo fazer uma pesquisa para identificar como a Engenharia de *Software* está sendo trabalhada na educação nos últimos anos frente ao crescente uso dos métodos ágeis no desenvolvimento de *software*. Ao concluir a pesquisa e realizar análise nos estudos encontrados podemos afirmar que nos últimos vinte anos, a utilização dos métodos ágeis foi adotada tanto nos cursos superiores na área da educação quanto na indústria de desenvolvimento de software. Em especial o XP (*Extremme Programming*), *Scrum* e o *Lean*, bem como a combinação de práticas ágeis de todos estes métodos são os resultados mais encontrados nesta pesquisa. É resultado também desta pesquisa uma análise do PPC e planos de ensino de disciplinas da linha de engenharia de software do curso de Análise e Desenvolvimento de Software do IFRS - Campus Restinga.

PALAVRAS-CHAVE: Métodos Ágeis, Engenharia de Software, Práticas Ágeis

ABSTRACT

After the Agile Manifesto, the use of Agile Methods became a reality in software development, making the academy adapt its curriculum to meet the needs of the industry. This Course Completion Work aims to do a research to identify how Software Engineering is being worked on in education in recent years in view of the increasing use of agile methods in software development. Upon concluding the research and performing analysis on the studies found, we can say that in the last twenty years, the use of agile methods has been adopted both in higher education courses in the area of education and in the software development industry. Especially XP (Extreme Programming), Scrum and Lean, as well as the combination of agile practices of all these methods are the most found results in this research. It is also a result of this research an analysis of the PPC and teaching plans of disciplines of the software engineering line of the course of Analysis and Software Development of IFRS - Campus Restinga.

KEYWORDS: *Agile Methods, Software Engineering, Agile Practices*

LISTA DE ILUSTRAÇÕES

Figura 1 - Etapas da Pesquisa.....	15
Figura 2 - Ciclo do <i>Scrum</i>	17
Figura 3 - Regras do <i>Extreme Programming</i>	19
Figura 4 - Uso geral de práticas ágeis.....	35

LISTA DE TABELAS

Tabela I - Melhores práticas aplicáveis aos alunos.....	30
Tabela II - Tendências SE comuns - lista final de SE <i>Trends</i>	32
Tabela III - Justificativas para os componentes mais importantes.....	37
Tabela IV - Subcomponentes totalmente recomendados.....	37
Tabela V - Subcomponentes fortemente recomendados.....	38
Tabela VI - Subcomponentes moderadamente recomendados.....	38
Tabela VII - Subcomponentes recomendados.....	39

SUMÁRIO

1. Introdução.....	12
1.1. Objetivos.....	14
1.2. Objetivo Específico.....	14
1.3. Etapas da Pesquisa.....	14
2. Metodologia Ágil.....	16
2.1. Scrum.....	16
2.2. <i>Extreme Programming (XP)</i>	18
2.3. <i>Lean Software Development</i>	19
3. Engenharia de <i>Software</i> na Educação.....	22
3.1. Estratégias de Ensino na Engenharia de <i>Software</i>	22
3.2. Práticas de Engenharia de <i>Software</i>	30
3.3. Análise do Contexto Atual da Engenharia <i>Software</i> na Educação.....	33
4. Engenharia de <i>Software</i> na Indústria.....	34
4.1 Análise do Contexto Atual da Engenharia <i>Software</i> na Indústria.....	42
5. Análise do PPC (Projeto Pedagógico de Curso) e Planos de Ensino das Disciplinas do Eixo de Engenharia de <i>Software</i> do IFRS - Restinga.....	43
6. Conclusão.....	45
Referências bibliográficas.....	47
Anexos.....	51

1. INTRODUÇÃO

Desde o surgimento da engenharia de *software* no final da década de 60, muitas práticas e técnicas surgiram e foram sendo aperfeiçoadas tanto pela academia quanto pela indústria de TI (Tecnologia da Informação). O objetivo é alcançar resultados que satisfaçam cada vez mais os clientes e que permitam ser mais assertivos quanto a prazo, custo, esforço, dentre outros aspectos relacionados ao desenvolvimento de *software*. Um importante marco na ES (Engenharia de *Software*) foi o manifesto ágil, o qual aconteceu em 2001, e consistiu na declaração de valores e princípios essenciais para o desenvolvimento de *software* baseado nos métodos ágeis (GAROUSI et al, 2016, DE O. MELO et al, 2013 e CICO et al, 2020).

Os métodos ágeis de desenvolvimento de *software* têm sido cada vez mais adotados em todo o mundo e se tornaram uma das principais abordagens de desenvolvimento de *software*. Isso porque, estes métodos podem ser aplicados a qualquer tipo de projeto de desenvolvimento de *software* com objetivo de aumentar a produtividade e encurtar os ciclos de entrega, melhorando a qualidade dos produtos. Atualmente os métodos ágeis mais usados na indústria são o *Scrum*, XP (*Extreme Programming*) e o *Lean Software Development* (DE O. MELO et al, 2013, CICO et al, 2020 e TAVARES et al, 2019).

O *Scrum* organiza o ciclo de vida do desenvolvimento de forma iterativa e incremental e é formado por um conjunto de boas práticas empregado no gerenciamento de projetos complexos em menor tempo e com o uso de menos recursos. Em um primeiro momento, teve utilização no desenvolvimento de softwares, mas sua dinâmica permite a aplicação em diferentes organizações que desejem aprimorar os processos de gestão de projetos (FIA, 2020).

O *Extreme Programming* (XP) tem muita semelhança com *Scrum* em termos de valores e modelo de desenvolvimento de projetos, ou seja, desenvolver *software* através de ciclos incrementais. No entanto, esses dois métodos também são complementares, visto que *Scrum* é mais como um *framework* gerencial. O XP desenvolve menos esses aspectos e foca mais em práticas de engenharia (MEDEIROS, 2013).

Por fim, o *Lean Software Development*, que em uma tradução literal significa “enxuto”, é uma filosofia de desenvolvimento baseada no sistema *Toyota* de produção. Ou seja, trata-se de um método que foca na eficiência de recursos e

institui o uso de nada além do que os recursos necessários para a realização de um determinado trabalho, etapa ou processo, evitando desperdícios (GDF, 2021).

O uso dos métodos descritos acima no desenvolvimento de *software* impactaram tanto o ensino de engenharia de *software* na academia, quanto o desenvolvimento de *software* por parte das indústrias de TI. No caso das universidades, houve uma grande transformação dos currículos dos cursos para acomodar essa nova forma de desenvolvimento de *software* (DE O. MELO et al 2013). E, uma vez que tecnologias, processos e práticas ágeis evoluem rapidamente, existe a constante necessidade de revisão e atualização dos cursos ligados a ES.

A indústria de desenvolvimento também teve uma grande transformação com o uso dos métodos ágeis. Isso porque, antes do desenvolvimento ágil, as metodologias tradicionais eram usadas, sendo estas metodologias mais rígidas e controladas. A adoção dos métodos ágeis trouxe muitas mudanças em práticas e técnicas usadas pela indústria, além de mudanças na cultura organizacional.

Para entender melhor os impactos dos métodos ágeis no ensino da Engenharia de *Software*, esta pesquisa visa realizar uma revisão da literatura especializada. O objetivo é descobrir como a Engenharia de *Software* está sendo trabalhada na educação nos últimos anos.

Para atender ao objetivo proposto foram pesquisados artigos científicos dos últimos vinte anos, mas mais especificamente artigos mais recentes para que pudessem ser encontradas quais as práticas de ensino estão sendo utilizadas e o que é necessário atualizar nos currículos de ES para acompanhar a evolução da indústria e assim formar profissionais mais preparados. Foi também realizada uma pesquisa sobre as práticas ágeis mais usadas na indústria com objetivo de correlacionar os resultados encontrados com os resultados encontrados na pesquisa sobre as práticas de ensino.

Será realizada ainda uma análise no PPC (Projeto Pedagógico de Curso) e planos de ensino das disciplinas do eixo de Engenharia de *Software* do curso de Análise e Desenvolvimento de Sistemas (ADS) do Câmpus Restinga do IFRS (Instituto Federal do Rio Grande do Sul). O objetivo desta análise é identificar possíveis oportunidades de melhoria para as disciplinas do curso baseado nos resultados encontrados nesta pesquisa.

Este Trabalho de Conclusão de Curso está organizado da seguinte forma: o capítulo 1 descreve esta introdução, os objetivos e a metodologia de pesquisa; o capítulo 2 descreve a Metodologia Ágil; o capítulo 3 descreve a Engenharia de *Software* no contexto da educação; o capítulo 4 apresenta as principais práticas de Engenharia de *Software* usadas na indústria; o capítulo 5 apresenta a análise do PPC e planos de ensino do curso de Análise e Desenvolvimento de Sistemas, seguido pelas conclusões descritas no capítulo 6.

1.1. Objetivos

Este trabalho tem como objetivo fazer uma pesquisa para identificar como a Engenharia de *Software* está sendo trabalhada na educação nos últimos anos frente ao crescente uso dos métodos ágeis no desenvolvimento de *software*.

1.2. Objetivo Específico

- Realizar levantamento bibliográfico de como a Engenharia de *Software* está sendo trabalhada na educação.
- Realizar levantamento bibliográfico de quais práticas ágeis estão sendo mais usadas na indústria.
- Realizar análise do que está sendo trabalhado na educação e as práticas da indústria de ES.
- Analisar o PPC (Projeto Pedagógico de Curso) e planos de ensino das disciplinas do eixo de Engenharia de *Software* do curso de Análise e Desenvolvimento de Sistemas do Câmpus Restinga.
- Sugerir oportunidades de melhoria para ementas de disciplinas e PPC do curso de Análise e Desenvolvimento de Sistemas do Câmpus Restinga.

1.3. Etapas da Pesquisa

A pesquisa foi realizada em várias etapas, as quais estão representadas pela Figura 1 e são descritas a seguir:

Etapa 1: nessa etapa realizou-se levantamento bibliográfico e estruturação da pesquisa.

Etapa 2: aqui analisaram-se os resultados da etapa 1 para a identificação dos artigos que seriam analisados em cada capítulo do trabalho.

Etapa 3: identificação de estratégias de ensino e práticas de Engenharia de Software utilizadas na academia e identificação das práticas de Engenharia de software utilizadas na indústria.

Etapa 4: a fase final constitui a análise do PPC do IFRS - Restinga e a correlação das práticas encontradas na academia e na indústria dos artigos analisados.

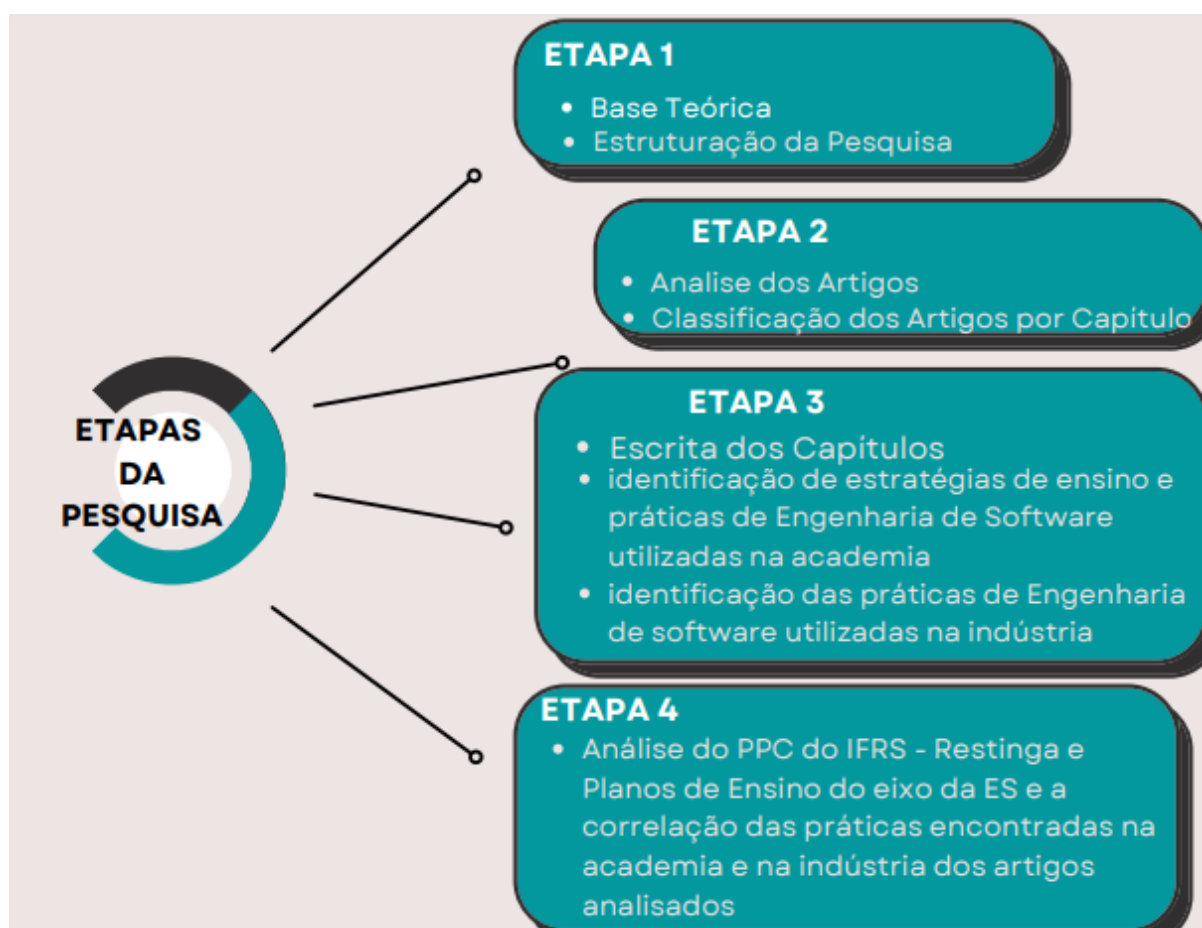


Figura 1 - Etapas da Pesquisa - Fonte: Próprio autor

2. METODOLOGIA ÁGIL

A metodologia ágil, ou *agile*, visa melhores resultados e aumento na produtividade. A busca pela padronização de processos e por práticas de excelência na gestão de projetos é constante em empresas que desejam a melhoria contínua de suas operações. A metodologia ágil surge como uma alternativa vantajosa, principalmente para organizações que atuam em setores ligados à tecnologia, engloba orientações e métodos distintos que contribuem para que equipes desenvolvam soluções mais eficientes e dinâmicas (TOTVS, 2021).

2.1. Scrum

Foi desenvolvido inicialmente por *Jeff Sutherland* e por sua equipe no início da década de 1990. O *Scrum* usa um conjunto de “padrões de processo de *software*”, que são adequados para projetos com prazos apertados e requisitos que mudam frequentemente (TOMÁS, 2009).

Scrum é uma estrutura leve que ajuda pessoas, equipes e organizações a gerar valor por meio de soluções adaptáveis para problemas complexos (scrum.org, 2022).

Em poucas palavras, *Scrum* exige que um *Scrum Master* promova um ambiente onde:

1. Um *Product Owner* ordena o trabalho para um problema complexo em um *Product Backlog*.
2. O Time *Scrum* transforma uma seleção do trabalho em um Incremento de valor durante um *Sprint*.
3. O Time *Scrum* e seus *stakeholders* inspecionam os resultados e se ajustam para o próximo *Sprint*.
4. *Repeat* (repita)

É o oposto de uma grande coleção de componentes obrigatórios entrelaçados. *Scrum* não é uma metodologia . *Scrum* implementa o método científico do empirismo. O *Scrum* substitui uma abordagem algorítmica programada por uma heurística, com respeito às pessoas e auto-organização para lidar com a

imprevisibilidade e resolver problemas complexos. O gráfico abaixo representa o *Scrum* em Ação conforme descrito por *Ken Schwaber* e *Jeff Sutherland*:

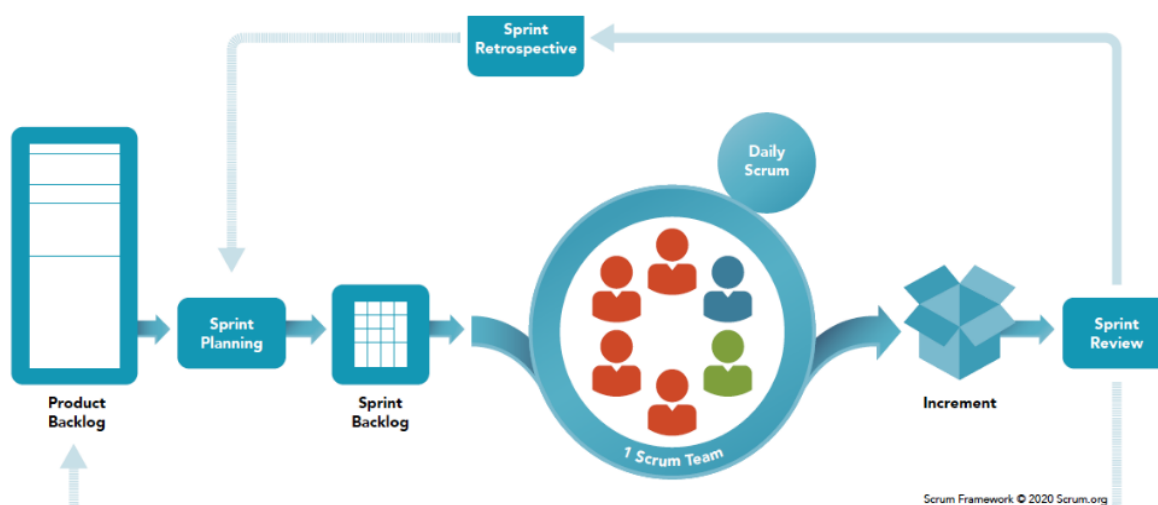


Figura 2 - Ciclo do *Scrum* - Fonte: Scrum Framework © 2020 Scrum.org

No *Scrum*, os projetos são divididos em ciclos (tipicamente mensais) chamados de *Sprints*. O Sprint representa um *Time Box* dentro do qual um conjunto de atividades deve ser executado. Metodologias ágeis de desenvolvimento de *software* são iterativas, ou seja, o trabalho é dividido em iterações, que são chamadas de *Sprints* no caso do *Scrum* (DESENVOLVIMENTO ÁGIL, 2013/2014).

As funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como *Product Backlog*. No início de cada *Sprint*, faz-se um *Sprint Planning Meeting*, ou seja, uma reunião de planejamento na qual o *Product Owner* prioriza os itens do *Product Backlog* e a equipe seleciona as atividades que ela será capaz de implementar durante o *Sprint* que se inicia. As tarefas alocadas em um *Sprint* são transferidas do *Product Backlog* para o *Sprint Backlog*.

A cada dia de uma *Sprint*, a equipe faz uma breve reunião (normalmente de manhã), chamada *Daily Scrum*. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia.

Ao final de um *Sprint*, a equipe apresenta as funcionalidades implementadas em uma *Sprint Review Meeting*. Finalmente, faz-se uma *Sprint Retrospective* e a equipe parte para o planejamento do próximo *Sprint*.

2.2. Extreme Programming (XP)

Criado em 1997 o XP por *Kent Beck*, o XP preconiza ciclos curtos que nos dá previsibilidade e redução de incertezas/riscos, Simplicidade e melhorias constantes de código (*refactoring*) para facilitar a mudança e Testes Automatizados e Integração Contínua para aumentar a confiança (MEDEIROS , 2013).

As práticas do XP são fundamentadas em valores. Entre os valores temos:

Comunicação: manter a comunicação seja entre desenvolvedores ou com os clientes. De preferência os clientes devem estar sempre presentes para criar Histórias de usuário e cliente on-site (CCC) ou ainda tirar dúvidas. Outra forma de comunicação no XP é a Programação em Pares, onde os desenvolvedores programam num mesmo computador, nesse formato de programação ambos estão constantemente se comunicando e trocando ideias. O Jogo do Planejamento (*planning poker*) também é outra forma de comunicação visto que a equipe de desenvolvimento está dando sua visão técnica, o cliente por sua vez está dando requisitos em prol do negócio e dando as prioridades. A comunicação ajuda na eliminação de documentos e favorece a comunicação face a face.

Simplicidade: é tentar fazer o mais simples possível e caso seja necessário faremos algo mais complexo amanhã. Muitas vezes algo é feito de forma completa e posteriormente não é mais sequer usado ou necessário. Aqui também temos a importância do coach que deve estar sempre verificando se a simplicidade está sempre sendo seguida nos projetos.

Feedback: é o valor primordial dentro do desenvolvimento ágil, ele possibilita que o *software* evolua. Os testes automatizados que permitem *feedback* rápido e respondem de forma imediata se aquilo que foi introduzido ainda está funcionando.

Coragem: XP diz que devemos ter coragem de sempre colocar o cliente a par do que está acontecendo. Entre aquilo que o XP considera que devemos ter coragem de fazer destacam-se:

- Acreditar na capacidade de reagir a mudanças;
- Trocar de paradigma;
- Aprender com os erros;
- Dar e receber *feedback* sem medo das consequências;
- Acreditar no *feedback* concreto (não na “teoria”);

- Fazer o que precisa ser feito;
- Jogar fora código ruim;
- Jogar fora protótipos criados para testar ideias.

Coach: é uma pessoa responsável por garantir a aderência a estes valores nas práticas. O *Coach* normalmente é uma pessoa experiente que também ajuda as equipes a implementarem o XP e monitorar se as coisas estão sendo bem seguidas.

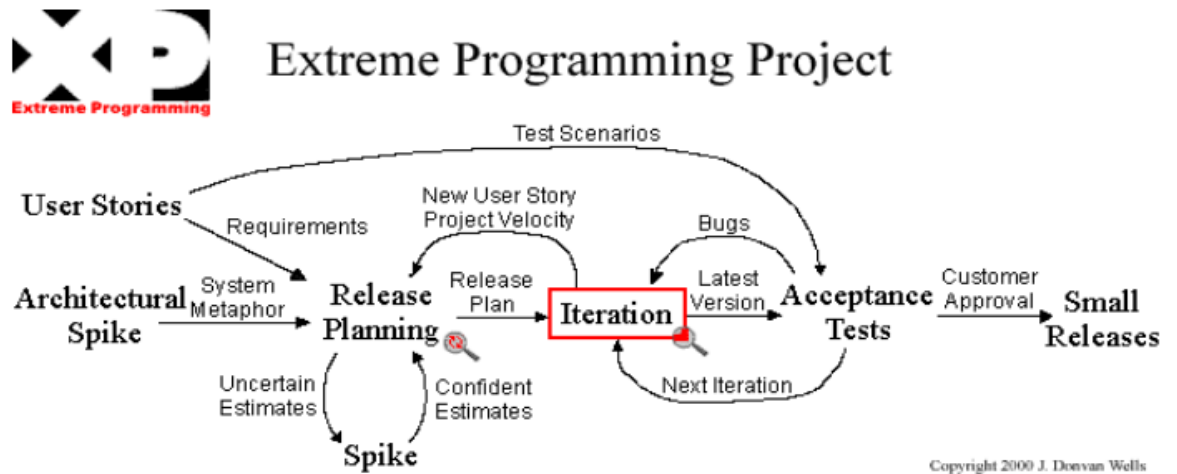


Figura 3 - Regras do Extreme Programming - Fonte: Copyright 2000 Don Wells

2.3 Lean Software Development

O conceito começou a ser utilizado na década de 1980 como “*Lean Manufacture*” (Manufatura Enxuta). Na época, carregava a filosofia de otimizar a produção de veículos da indústria japonesa. No entanto, foi o professor *James P. Womack*, do MIT (*Massachusetts Institute of Technology*) que tornou a filosofia *lean* conhecida mundialmente, após a publicação de seu livro *A máquina que mudou o mundo*, em 1990, – escrito em conjunto com Daniel T. Jones e Daniel Roos. Nesse livro, *Womack* detalhou seu estudo sobre a indústria automobilística (EQUIPE RUNRUN.IT, 2013).

Lean Software Development trás os conceitos de Lean para o universo do desenvolvimento de software, para que através da aplicação dos mesmos princípios seja possível eliminar desperdícios e alcançar melhores resultados. Ela foca na eliminação de desperdícios, aumento da velocidade de processos e na excelência em qualidade. Implementar Lean permite que uma organização diminua seus estoques, maximize o uso de trabalhadores generalistas (ou seja, que possuem

muitas habilidades) e produza de acordo com a demanda. Lean também é conhecido como TPS, Toyota Production System ou Sistema de Produção Toyota (REVISTA JAVA MAGAZINE 81, 2010).

Lean oferece um conjunto de princípios que podem ser utilizados por organizações para adaptar ferramentas, técnicas e métodos a seus contextos e capacidades específicas.

O termo Lean Software Development teve sua origem em 2003 na publicação de um livro de mesmo nome escrito por Tom e Mary Poppendieck. Neste trabalho os autores apresentam como aplicar princípios de Lean ao desenvolvimento de software.

Princípios da metodologia *Lean* (FONTES, 2018)

São sete princípios presentes nessa filosofia e que regem a sua aplicação:

1. Processamento impróprio

É o principal foco dentro da Metodologia *Lean*. É necessário eliminar os 8 desperdícios que estão presentes nos processos para aumentar a produção. São eles: transporte, estoque, movimentação desnecessária, espera, produção excessiva, processamento impróprio, defeitos e conhecimento (mal aproveitado).

2. Fortalecer a equipe

É essencial mostrar a todos os funcionários qual a sua importância e como a sua contribuição individual é crucial para o projeto e para a empresa. Assim, é possível engajar a equipe e aumentar a produtividade.

3. Amplificar o conhecimento

Do mesmo modo que o incentivo deve ser aplicado em toda a equipe, o conhecimento também precisa ser compartilhado para que todos estejam alinhados e com consciência dos conceitos e métodos que serão utilizados.

4. Construir qualidade

A qualidade é um dos principais pontos dentro da metodologia *Lean*. Apesar de ser um foco, ela acaba se tornando uma consequência dos outros princípios. Pense que, se há eliminação de desperdícios, redução do tempo de cada processo e valor agregado apenas nas atividades que visam a satisfação do cliente, o resultado é a qualidade.

5. Entregas rápidas

O *Lean* busca a redução do *Lead Time*, ou seja, o tempo entre o pedido feito pelo cliente e a entrega do mesmo. Nesse ponto é importante a verificação dos gargalos da produção, para que seja possível solucioná-los e conseqüentemente chegar ao final do processo mais rápido.

6. Adiar decisões

Nesse caso, adiar decisões não quer dizer procrastinar. Esse princípio está ligado à flexibilização da produção, onde mudanças podem acontecer no meio do processo. Assim, não há um fechamento em apenas uma alternativa que não pode ser alterada.

7. Otimizar o todo

Com a colaboração de todos empenhados em melhorar cada etapa, não separadamente, de forma coesa e vista de maneira geral, se chega na tão sonhada otimização do processo. Dessa forma, é possível que a produção transcorra de forma alinhada e eficiente.

Os princípios apresentados pela filosofia *Lean* podem ser adequados em diversas aplicações:

Lean Manufacturing - O *Lean Manufacturing*, ou "Manufatura Enxuta", dentro de uma organização, o *Lean* busca utilizar o mínimo de recursos, reduzindo ou até mesmo eliminando atividades que não agregam valor, através da identificação dos desperdícios, visando melhorar o uso dos indicadores de qualidade, custo e entrega.

Lean Office - O *Lean Office* utiliza os conceitos da filosofia *Lean* no ambiente de escritório ou áreas administrativas de uma empresa. Sendo assim, a metodologia enxuta não se aplica mais em atividades manufatureiras e físicas, nesse caso o fluxo de valor passa a ser o fluxo de informações.

Lean Construction - No caso do *Lean Construction*, a aplicação do *Lean* se dá na construção civil, onde o chão de fábrica é trocado pelo canteiro de obras. Porém, na construção civil o local de trabalho está em constante mudança, logo precisa adaptar o pensamento enxuto para estas e outras peculiaridades. O *Lean Construction* busca a rapidez da entrega da obra, a redução de custos e diminuição dos prejuízos de uma construção.

Lean Healthcare - A filosofia *Lean* é aqui aplicada na área da saúde, utilizando seus conceitos para melhorar o atendimento aos pacientes, diminuir

desperdícios e ainda buscar a melhoria contínua de todos os funcionários que participam do processo. Além disso, o *Lean Healthcare* também auxilia na organização e administração das casas de saúde.

Lean Startup - O *Lean Startup* busca uma aplicação mais abrangente da filosofia *Lean*, aplicando suas ferramentas e conceitos no processo empreendedor. As táticas de venda são otimizadas nesta aplicação, onde também é muito utilizado o MVP (*Minimum Viable Product*). Nesse caso, a redução de desperdícios gira em torno do desenvolvimento da clientela, investindo o tempo em clientes reais e reduzindo ao máximo o número de falsas suposições o quanto antes.

Lean Logistic - O objetivo da Logística *Lean* é aplicar o pensamento enxuto para descartar os desperdícios que aparecem nas operações logísticas e conseqüentemente reduzir custos. Uma das ferramentas que auxiliam nesse processo é o Mapa do Fluxo de Valor, que aponta para o gestor quais atividades agregam valor à operação.

3. ENGENHARIA DE SOFTWARE NA EDUCAÇÃO

Este capítulo analisa a engenharia de *software* no contexto da educação. Inicialmente, na seção 3.1 serão descritas quais as estratégias de ensino para engenharia de *software* são mais adequadas. Isto é importante, pois a principal lacuna apontada pelos estudos é que a maioria dos currículos prevê somente atividades teóricas feitas em sala de aula e não prepara os alunos com aulas práticas vivenciando situações reais de desenvolvimento de *software*. Em seguida, a seção 3.2 levanta quais são as práticas de engenharia de *software* que estão sendo ensinadas na academia. O objetivo é identificar nos estudos quais práticas são essenciais em um currículo de engenharia de *software*.

3.1. Estratégias de Ensino na Engenharia de Software

A indústria de *software* e a academia de engenharia de *software* (ES) são duas grandes comunidades. No entanto, infelizmente, o nível de colaboração conjunta indústria-academia em ES ainda é relativamente baixo, em comparação

com a quantidade de atividade em cada uma das duas comunidades. Parece que os dois 'campos' mostram apenas interesse/motivação limitados para colaborar um com o outro. Muitos pesquisadores e profissionais escreveram sobre os desafios, padrões de sucesso (o que fazer, ou seja, como colaborar) e antipadrões (o que não fazer) para colaborações indústria-academia (GAROUSI et al, 2016).

Pesquisadores sugerem o uso de laboratórios que trabalhem o desenvolvimento de *software* com todos os estágios, usando métodos ágeis, para que os alunos vivenciem na prática situações reais. Esta experiência poderá envolver a interdisciplinaridade para que o ensino da Engenharia de *Software* possa ser trabalhado durante um período maior do curso, fazendo com que tenhamos profissionais mais preparados.

Rayford et al. (2006) sugerem alguns requisitos que são necessários para a indústria a partir dos graduados. Entre os requisitos estão a capacidade de trabalhar sob estresse, boas habilidades de comunicação e capacidade de acompanhar processos, uma sólida ética de trabalho e habilidades de gerenciamento de projetos. Esses requisitos são o que falta à maioria dos formandos (ALMI, 2011).

Para contornar o problema, que é a falta de qualificação dos graduados, algumas empresas precisam fornecer centros de treinamento (internos e externos) aos graduados (G. TARAN et al, 2007 e ALMI 2011).

A maioria dos graduados ainda não está pronta para enfrentar o ambiente real de trabalho, a menos que participe de alguns cursos extras para aprimorar suas habilidades sociais, gerenciais e técnicas antes de buscarem seu primeiro emprego após a graduação (ALMI, 2011).

Segundo Dagnino (2014), recém-formados que começam a trabalhar na indústria costumam se surpreender com a quantidade de elementos da área de Engenharia de *Software* que não foram expostos durante seus estudos acadêmicos. Uma maneira de preencher essa lacuna é desenvolver um método que inclua não apenas o ensino da teoria da engenharia de *software*, mas também permita que os alunos pratiquem os princípios em um ambiente de indústria simulado e desenvolvam habilidades úteis para a disciplina.

No curso de Graduação em Engenharia de *Software* (CSC 510) no Departamento de Ciência da Computação da Universidade Estadual da Carolina do Norte, é utilizado um método para ensinar aos alunos de Engenharia de *Software*

habilidades importantes para seu futuro ambiente profissional (DAGNINO, 2014), conforme relatamos a seguir:

O método tem quatro elementos principais: abordagem de ensino utilizada para ministrar o curso; artefatos que os alunos usam para aprender o assunto; atividades que os alunos desenvolvem ao longo do curso para a aprendizagem da matéria; disciplinas adicionais não tradicionais relevantes para a prática da Engenharia de *Software* que são ensinadas aos alunos.

Detalhando a forma de aplicação do projeto da Universidade Estadual da Carolina do Norte citaremos em tópicos os métodos:

- Um elemento importante no ensino da Engenharia de *Software* é a experiência do instrutor, o que agrega ao curso além dos conceitos acadêmicos, a forma como são colocados em prática na indústria: (a) desempenhar diferentes papéis no setor durante os exercícios de aula e atribuições progressivas, como cliente, gerente de projeto, gerente de produto; (b) orientar os alunos em suas atividades de aprendizagem e também na dramatização; (c) desenvolver e fornecer modelos orientados para a indústria; (d) fornecer “histórias” da indústria; (e) e trazer uma compreensão sobre como a indústria de *software* funciona e valores.
- São usadas as áreas de processo do CMMI (*Capability Maturity Model Integration*) Nível 2 e Nível 3 (detalhados na seção 3.2),
- Exemplos da indústria e histórias originadas de experiências diretas que o instrutor viveu diretamente relacionadas aos tópicos sendo ensinados durante o curso.
- Na camada de artefatos de aprendizagem, os alunos precisam aprender que os modelos e ferramentas são essenciais na indústria, pois devem incorporar os processos de desenvolvimento institucionalizados. Os modelos e ferramentas devem ser simples de usar e compreender, agregar valor aos usuários e ajudar a garantir a alta qualidade dos produtos de trabalho. Também é experiência do autor que esses modelos e ferramentas não devem ser vistos como um “fardo” para os usuários e, portanto, devem envolver os conceitos de desenvolvimento ágil de *software*. É importante que os alunos de Engenharia de *Software* sejam expostos a esses modelos e

ferramentas e os experimentem diretamente, para que se sintam confortáveis em usá-los na indústria após a graduação.

- Exercícios de interpretação de papéis da indústria: os exercícios de dramatização ajudam a simular um ambiente industrial e permitem que os alunos vivenciem diretamente as situações que enfrentarão em suas carreiras futuras. Os exercícios de dramatização precisam ser ilustrativos e curtos o suficiente para fazer parte da instrução teórica durante a aula. Os exercícios de dramatização incluem: (a) estimativa de *software*; (b) revisões por pares; (c) identificação e mitigação de riscos; (d) teste de caixa preta; (e) avaliação do processo de *software*; e (f) gerenciar a mudança organizacional. Misturar a teoria e fazer os alunos se dividirem em grupos de trabalho para ter experiências práticas nesses exercícios é muito importante.
- Tarefas Progressivas: são tarefas que se complementam e são feitas individualmente por cada aluno. O objetivo dessas tarefas é que os alunos comecem individualmente um projeto desde os estágios iniciais do ciclo de vida de desenvolvimento de *software* e concluam a tarefa em que uma arquitetura parcial de um sistema de *software* é desenvolvida. Em um semestre, o número ideal de atribuições progressivas é quatro. No início do semestre, os alunos selecionam um dos três sistemas de *software* do mundo real em potencial para construir nas atribuições progressivas e quatro atribuições são definidas. Em primeiro lugar, a partir de uma definição de alto nível do sistema (o instrutor é o patrocinador/cliente do projeto), o objetivo da primeira tarefa é desenvolver um caso de negócio inicial associado ao *software* a ser construído, definir uma lista parcial de casos de uso, recursos e requisitos de mercado. Na segunda tarefa, os alunos desenvolvem requisitos técnicos e de produto a partir dos requisitos de mercado definidos. Na terceira tarefa, os alunos desenvolvem um plano de projeto, com estimativas, recursos, prazos, riscos, EAP(Estrutura Analítica do Projeto) e habilidades necessárias para realizar o projeto. No trabalho final, os alunos desenvolvem a arquitetura definindo os cenários de qualidade a partir dos drivers de negócios e requisitos definidos.

- Demonstrador de projeto de equipe e arquitetura: Um projeto de equipe é formado por três a quatro alunos. Cada equipe utiliza os materiais desenvolvidos nas atribuições progressivas individuais e decide quais elementos de suas atribuições são considerados para implementar certas funcionalidades do *software*, utilizando a arquitetura e tecnologia de sua escolha. O objetivo é que a equipe trabalhe em conjunto para desenvolver e implementar um demonstrador de sua arquitetura de sistema, priorizando cenários de arquitetura e atributos de qualidade definidos. Além disso, a equipe deve desenvolver casos de teste que são utilizados para verificar a arquitetura e garantir que os cenários de atributos de qualidade sejam implementados corretamente (DAGNINO, 2014).
- Habilidades de apresentação e facilitação: Descendo para os tópicos não tradicionais de Engenharia de *Software*, os alunos são ensinados ao longo da apresentação do curso e habilidades de facilitação por meio de exercícios em classe, exercícios de simulação e, por fim, as apresentações finais do projeto. Quando os alunos começam suas carreiras profissionais, eles descobrem que as habilidades de apresentação e facilitação são essenciais para o sucesso.
- Gerenciando Mudanças Organizacionais: Para uma organização ter uma cultura de melhoria contínua de processo bem-sucedida, é necessário gerenciar a mudança organizacional de forma muito eficaz. É importante que os alunos aprendam modelos para avaliar a prontidão de uma organização para implementar mudanças em um determinado momento. Os alunos então aprendem os diferentes papéis e seus comportamentos em uma situação de mudança que o patrocinador, o agente de mudança, as pessoas afetadas pela mudança, os defensores da mudança e os detratores da mudança desempenham. Tópicos como resistência aberta e encoberta são discutidos. Além disso, são discutidas estratégias sobre como melhorar a prontidão da organização para abraçar a mudança de forma eficaz e manter um ambiente saudável de melhoria contínua de processos.

- Engenharia de *software* e valor comercial: Todas as organizações de desenvolvimento de *software* são impulsionadas pelo aumento dos lucros e redução dos custos. Conseqüentemente, todos os processos, avaliações, programas de melhoria, desenvolvimento e atividades da Engenharia de *Software* são orientados por análises financeiras. É essencial que os alunos de engenharia de software entendam as implicações financeiras da disciplina.
- Palestrantes convidados da indústria: O autor observou que os alunos adquirem uma experiência inestimável e apreciam o fato de os palestrantes convidados da indústria fazerem apresentações ao longo do curso. Isso permite que os alunos tenham uma variedade de pontos de vista e enriqueçam suas experiências de aprendizagem. Ter de três a quatro palestrantes convidados da indústria é um bom número em um curso semestral.

No Brasil, temos algumas academias que já utilizam o uso de laboratórios de metodologias ágeis nos cursos de graduação e pós-graduação e que relatam experiências positivas quanto ao aprendizado e preparo dos alunos (DE O. MELO et al, 2013). Entre elas podemos citar: USP, UFRJ, UFPE, UFRPE e PUCRS.

O Laboratório XP do IME-USP (Instituto de Matemática e Estatística da Universidade de São Paulo) teve sua primeira edição em 2001, com um grupo pequeno de alunos e foi evoluindo ao longo dos anos. Para criar um ambiente mais próximo do real, os clientes do curso IME-USP são escolhidos a partir de solicitações de universidades ou projetos de código aberto. Todos os sistemas desenvolvidos estão disponíveis como software livre. O curso começa com três semanas de aulas teóricas, quando são introduzidos os fundamentos dos métodos ágeis e XP; então, os alunos podem escolher os projetos em que desejam trabalhar. Normalmente, cada projeto recebe de quatro a oito participantes. Se não houver aluno experiente no grupo, um treinador é eleito entre os voluntários. Os grupos sem *coach* experiente recebem mais atenção dos meta-coaches. O curso IME-USP exige 8 horas semanais de dedicação dos alunos. As práticas iniciais adotadas no início dos projetos são as 12 da primeira edição do livro XP; além disso, reuniões diárias e espaços de trabalho informativos também são usados. Todas as equipes devem realizar retrospectivas ao final de cada iteração. Uma vez por semana, há uma reunião envolvendo coach, meta-coach e o professor para revisões do projeto,

compartilhamento de problemas ou soluções e solicitação de apoio. O professor e os *meta-coach* monitoram as práticas ágeis das equipes, avaliam seus espaços de trabalho informativos e o progresso, e também coletam *feedback* dos alunos sobre o curso para adaptá-lo de acordo com as necessidades de aprendizagem dos alunos. Para avaliar os alunos, é calculada uma média de notas, frequência, participação pró-ativa, rastreamento, satisfação do cliente, junto com a avaliação pessoal, do *coach* e do *meta-coach* (DE O. MELO et al, 2013).

Em 2006, à medida que os métodos ágeis começaram a ganhar ampla aceitação, alguns professores e alunos do IME-USP decidiram oferecer um curso de verão para divulgar essas ideias além dos limites da universidade. Foi um curso teórico de 20 horas, distribuídas em 5 dias, com dois instrutores por dia. O resultado foi um sucesso e todo o material de ensino foi publicado no site da Agilcoop para ser usado gratuitamente sob uma licença *Creative Commons*. Nos 3 anos seguintes, a equipe de instrutores continuou a oferecer o curso, adicionando tópicos como um Laboratório XP para oferecer uma abordagem mais prática e um curso de teste para estudar esse assunto mais profundamente. Nesse período, mais de 200 pessoas participaram do curso teórico, mais de 60 participaram do curso prático e cerca de 50 estavam no curso de testes. O máximo de esses 300 participantes eram do setor e ajudaram a disseminar as práticas ágeis em suas empresas (DE O. MELO et al, 2013).

Na Universidade Federal do Rio de Janeiro (UFRJ) foi realizada a iniciativa no ensino do XP a partir do segundo semestre de 2002. Onde cada aula teve duas partes: a primeira teórica em que os alunos deveriam responder a perguntas oralmente com base no material fornecido anteriormente. Para a parte prática, ao invés de ter um projeto para cada equipe durante todo o curso, vários exercícios curtos em cada prática foram propostos. A motivação era reforçar o aprendizado de cada prática, sempre praticando a programação em pares. A cada dia, os pares eram escolhidos aleatoriamente. As notas foram atribuídas principalmente com base na assiduidade e nas respostas (DE O. MELO et al, 2013).

Na Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), por exemplo, no sul do país, há um programa de três cursos sobre métodos ágeis que é oferecido todos os anos pela Faculdade de Informática. Os cursos envolvem um curso introdutório, gerenciamento ágil de projetos com *Scrum* e análise ágil de

negócios. Mais recentemente, a PUCRS se envolveu em outra iniciativa, que envolve a criação de um laboratório ágil financiado pelo CNPq (Conselho Nacional de Pesquisas - agência de fomento brasileira) para desenvolver equipes de desenvolvimento de software de alta performance, com base em métodos ágeis (DE O. MELO et al, 2013).

O curso de Bacharelado em Engenharia de *Software* da Faculdade de Informática da PUCRS (YAMAGUTI, 2017), foi criado, em 2015, com foco na formação de profissionais mais capacitados e qualificados para atuação nas empresas de TI. Desde a sua concepção este curso foi criado buscando formas e modelos que permitissem a formação integrada de conhecimentos que não fosse fundamentada apenas em uma matriz curricular de disciplinas, utilizando uma abordagem de cunho construtivista. A AGES - Agência Experimental de Engenharia de *Software*, é um espaço que foi criado como componente curricular do curso de Engenharia de Software. Foi projetada para se integrar na matriz curricular do curso de Engenharia de Software, no qual todos os seus alunos, devem participar de atividades da Agência. Mais especificamente, todo aluno deve realizar 4 módulos previstos no projeto pedagógico do curso (AGES I, AGES II, AGES III e AGES IV). AGES I ocorre no 2º semestre do curso de ES, enquanto que AGES II no 4º semestre, AGES III no 6º e AGES IV no 7º semestre do curso. A cada semestre são desenvolvidos vários projetos que envolvem alunos, professores-orientadores e equipe técnica. Cada time de um projeto envolve alunos de diferentes módulos. Desta forma, há alunos de diferentes níveis (semestres) interagindo em cada time, o que permite a troca de informações e experiências entre os próprios alunos.

Após realização de todos os módulos na Agência, o aluno deve escrever e apresentar a uma banca de avaliação, com supervisão de um professor orientador, um memorial descritivo do seu portfólio realizado na Agência, em formato técnico a ser seguido. Neste memorial, o aluno contextualiza as atividades relatadas no portfólio, com a devida fundamentação teórica baseada em seus estudos nas disciplinas do Curso, e suas conclusões, contribuições e lições aprendidas a partir dessas vivências. Esta apresentação tem como objetivo a avaliação do conjunto de atividades e conhecimentos do aluno, considerando as competências e habilidades esperados de um egresso do curso de Engenharia de Software.

3.2. Práticas de Engenharia de Software

Maíra Marques e Javier Robledo (2018), realizaram revisão sistemática da literatura para descobrir quais são as melhores práticas de engenharia de *software* que estão sendo ensinadas aos alunos na academia. Onde esperava-se mostrar quais são as melhores práticas que estão sendo ensinadas e como, para que outros instrutores e o corpo docente possam avaliar o que escolher e nivelar seus cursos na medida do que está sendo usado e já foi testado. As melhores práticas foram divididas em melhores práticas instrucionais (metodologia de ensino) e melhores práticas aplicáveis aos alunos. Na Tabela I veja as melhores práticas aplicáveis aos alunos. Práticas relatadas terem sido ensinadas em mais de um estudo primário (estão em negrito na Tabela I).

Tabela I - Melhores práticas aplicáveis aos alunos

PS	Ano	Melhores práticas relatadas (aplicável aos alunos)
[31]	1996	práticas reflexivas , uso de inspeções , métodos formais, processo de integração , mentorias, satisfação do cliente, acampamento de treinamento, requisitos , Processo de <i>Software</i> Pessoal (PSP)
[32]	1999	Aprendendo com os erros (project postmortems)
[11]	2001	prototipagem, Garantia da Qualidade
[2]	2003	inspeção de código
[28]	2005	práticas de <i>Extremme Programming</i> (Planejamento(planning game), pequenos lançamentos, metáfora, <i>design</i> simples, testes , reestruturação (<i>refactoring</i>), programação em par , propriedade coletiva, integração contínua , semana de quarenta horas (<i>forty-hour week</i>), cliente no local e padrões de codificação)
[33]	2007	Contatos, reciprocidade, <i>feedback</i> , tarefas de tempo, respeito(<i>respect</i>)
[15]	2009	gerenciamento de requisitos , análise e design orientado a objetos, gerenciamento de riscos, gestão da qualidade , revisões de pares, gerenciamento de configurações, e engenharia de software baseada em valor
[22]	2012	reuniões diárias online, líder de grupo, desenvolvimento iterativo e incremental, redistribuição de atividades, controle de versão, grupos auto-gerenciáveis, programação em par online
[10]	2013	preparação, reflexão
[17]	2015	análise de causa raiz (<i>root cause analysis</i>)

[27]	2015	requisitos, planejamento, design , avaliação, realização, controle, planejamento do projeto
[20]	2015	requisitos , análise, design, testes , implementação, manutenção, segurança
[19]	2016	métrica de conformidade
[6]	2016	Padrões de <i>design</i>
[5]	2017	análise estática, teste de unidade, teste de integração, construção contínua

PS - Estudos Primários - Fonte: Marques et al. (2018).

Foi estabelecido que existem algumas práticas que foram relatadas como sendo utilizadas em mais de um estudo primário, que pode ser considerado um bom indicador de que são realmente as "melhores práticas" de engenharia de *software* que devem ser levadas em consideração dentro do contexto do ensino de engenharia de *software*, são eles: integração, qualidade, reflexão, requisitos, programação em pares, testes/inspeções, projeto, planejamento, teste e integração contínua (construção) (MARQUES et al., 2018).

Dagnino (2014), sugere no curso de Graduação em Engenharia de *Software* (CSC 510) no Departamento de Ciência da Computação da Universidade Estadual da Carolina do Norte, que as áreas de processo do CMMI Nível 2 são fundamentais em uma organização de desenvolvimento de *software* para institucionalizar um processo de desenvolvimento gerenciado. Essas áreas de processo incluem planejamento de projeto (PP), monitoramento e controle de projeto (PMC), gerenciamento de requisitos (REQM), gerenciamento de configuração (CM), garantia de qualidade de produção e processo (PPQA), gerenciamento de contrato de fornecedor (SAM) e medição e análise (MA). Além das áreas de processo CMMI Nível 3 que também precisam ser incluídas no currículo incluem desenvolvimento de requisitos (RD), gerenciamento de requisitos (RM), solução técnica (TS), integração de produto (PI), verificação (VER) e validação (VAL). Fornecer detalhes sobre essas áreas de processo e explicar como elas são implementadas na indústria é muito importante para que os alunos obtenham uma apreciação sobre como as indústrias de software maduras implementam essas áreas de processo.

Cico et al. (2020), relatam em estudo um mapeamento sistemático de artigos empíricos de Educação em Engenharia de *Software* escritos sobre Tendências de Engenharia de Software, no qual identificaram como a tendência em

Engenharia de *Software* mais popular em educação, denominada *Agile Software Development*, seguida de Implementação de *software*, usabilidade e valor. Outras tendências da Engenharia de *Software*, como *Lean Software Startup* e *Global Software Engineering*, apresentam uma série menor de estudos da Engenharia de *Software* na Educação, mas continuam apresentando crescimento nos últimos cinco anos. Assim, refletindo uma lacuna que necessita atenção de pesquisadores e educadores. A tendência em Engenharia de *Software System of Systems* é pouco explorada no contexto de Engenharia de *Software* na Educação. Também descobriram que as práticas do *Scrum* são as metodologias mais comuns. Esforços são feitos para combinar métodos *Agile* com *Lean Software Development* e GSE.

A integração da tendência em Engenharia de *Software* é abordada principalmente por abordagens de aprendizagem baseadas em projeto na Engenharia de *Software* na Educação. Outras abordagens de aprendizagem em potencial, ou seja, aprendizagem gamificada e aprendizagem híbrida, recebem um mínimo de exploração. Observamos que a participação real dos *stakeholders* industriais na Engenharia de *Software* na Educação permanece limitada; impactando a tendência de adoção de novas Tendências de Engenharia de *Software*, que podem levar a Engenharia de *Software* na Educação a um período de estagnação (CICO et al,2020).

Tabela II - Tendências ES comuns - lista final de SE Trends

Categoria	Propriedades
<i>Agile Software Development</i> (Desenvolvimento Ágil de <i>Software</i>)	Práticas ágeis, <i>Lean</i> e <i>Agile</i> , prototipagem rápida, Integração de software empresarial, DevOps , integração contínua de software
Implementação de <i>Software</i> Usabilidade e valor	<i>Software</i> centrado no usuário , usabilidade de <i>software</i> , requisitos não funcionais de <i>software</i> , valor do <i>software</i> , desenvolvimento de funcionalidade
Engenharia de <i>Software</i> Global	Engenharia de <i>Software</i> Global, ambiente multinacional vs. local, <i>cross-site</i> , desenvolvimento de <i>software</i> no mesmo <i>site</i> (<i>same-site</i>) , desenvolvimento distribuído
<i>Lean Software Startup</i>	Práticas inovadoras na indústria por meio de <i>startups</i> de tecnologia, <i>Lean Software Startups</i> , BizOps, <i>Tech Startups</i> , <i>Software Startups</i>
<i>System of Systems</i>	Sistemas de nuvem, sistemas <i>mobile</i> , <i>System of Systems</i> ,

Durante a pesquisa ficou claro que além do ensino de toda a parte teórica, é muito importante trabalhar as habilidades dos alunos em equipe para que tenham um maior preparo para o ingresso na indústria de software.

3.3. Análise do Contexto Atual da Engenharia Software na Educação

Analisando os artigos encontrados, a maior parte dos estudos falavam das habilidades necessárias para os graduados, além das metodologias ágeis já conhecidas como *Scrum*, *XP* e *Lean*. Poucos artigos foram encontrados relatando quais as práticas efetivamente utilizadas nas academias, o que acabou dificultando o mapeamento que inicialmente foi planejado.

Nos artigos analisados, em sua maioria, os pesquisadores sugeriram o uso de laboratórios que trabalhem o desenvolvimento de *software* com todos os estágios, usando métodos ágeis, para que os alunos vivenciem na prática situações reais.

Tendo em vista, que cada disciplina de ES tem o período de um semestre, sendo trabalhada de forma individual sugere-se envolver a interdisciplinaridade para que o ensino da Engenharia de *Software* possa ser trabalhado durante um período maior do curso, fazendo com que tenhamos profissionais mais preparados.

Além da capacidade técnica, os graduados devem ter os seguintes requisitos: capacidade de trabalhar sob estresse, boas habilidades de comunicação e capacidade de acompanhar processos, uma sólida ética de trabalho e habilidades de gerenciamento de projetos. Esses requisitos são de extrema importância para o bom funcionamento do time, pois estarão convivendo com pessoas de diferentes vivências, o que poderá dificultar a comunicação, esse pode ser o requisito de maior importância para desenvolvimento em equipe.

Com referência às metodologias ágeis as mesmas são trabalhadas de forma individual, enquanto que foram encontrados estudos da combinação de práticas ágeis. Sem dúvida alguma, as metodologias devem ser ensinadas conforme a literatura, mas essas combinações devem ser evidenciadas para que o

aluno possa ter o conhecimento da possibilidade de unir práticas diferentes conforme a área de atuação.

4. ENGENHARIA DE SOFTWARE NA INDÚSTRIA

Este capítulo analisa a engenharia de *software* no contexto da indústria. O objetivo é identificar nos estudos quais práticas e técnicas estão sendo utilizadas na indústria de desenvolvimento de *software*.

Segundo Diebold e Dahlem (2014), o desenvolvimento ágil de *software* tem sido cada vez mais adotado nas últimas duas décadas. No entanto, muitos estudos mostram que o uso de métodos ágeis conforme definido na literatura não funciona muito bem. Assim, as empresas adaptam esses métodos usando apenas partes deles (chamadas práticas ágeis). Foi realizado um mapeamento de estudos empíricos utilizando práticas ágeis na indústria no qual os resultados mostram que existem práticas que são utilizadas com mais frequência e que o domínio e o processo também influenciam na aplicação de diferentes práticas.

Os métodos mais usados para desenvolvimento ágil são *Scrum* e *Extreme Programming (XP)*. Porém, são poucas as empresas que realmente aplicam esses métodos conforme descrito na literatura. A maioria das empresas adapta esses métodos devido a vários problemas, como complexidade encontrada durante a introdução ou mudança para o desenvolvimento ágil. Na maioria das vezes, eles omitem partes específicas do método ágil original, as alteram ou as substituem por aspectos tradicionais. A adaptação mais proeminente é o chamado método “*ScrumBut*”, que usa o *Scrum* até certo ponto (DIEBOLD et al,2014).

Para obter um conjunto de práticas ágeis universais, analisamos os métodos ágeis mais comuns, extraímos as práticas e as agrupamos sob um nome comum. Isso resultou na seguinte lista de práticas ágeis universais (DIEBOLD et al.,2014):

1. Verificação de qualidade
2. Refatoração
3. Envolvimento do cliente
4. Equipes comunicativas não vinculadas
5. Prática de validação

6. Ciclo de aprendizagem
7. Revisão do resultado
8. Reunião de planejamento
9. *Time boxing*
10. Conhecimento comum
11. Monitoramento do progresso
12. Visão do produto
13. Evolução e especificação hierárquica
14. Integração/implantação contínua
15. Entrega de lançamentos frequentes
16. Pequenas equipes multifuncionais
17. Discussão diária
18. Análise de especificação contínua

Os resultados gerais deste estudo são apresentados no gráfico mostrado na Figura 4, que mostra o uso (total e parcial) e o não uso de práticas ágeis com base em todos os 68 projetos dos 24 estudos. O eixo y apresenta as diferentes práticas ágeis universais, com cada prática sendo codificada com três barras para uso total (preto), uso parcial (cinza) e não uso (branco). O eixo x do gráfico mostra a frequência com que as diferentes práticas ágeis foram encontradas nos artigos (DIEBOLD et al.,2014).

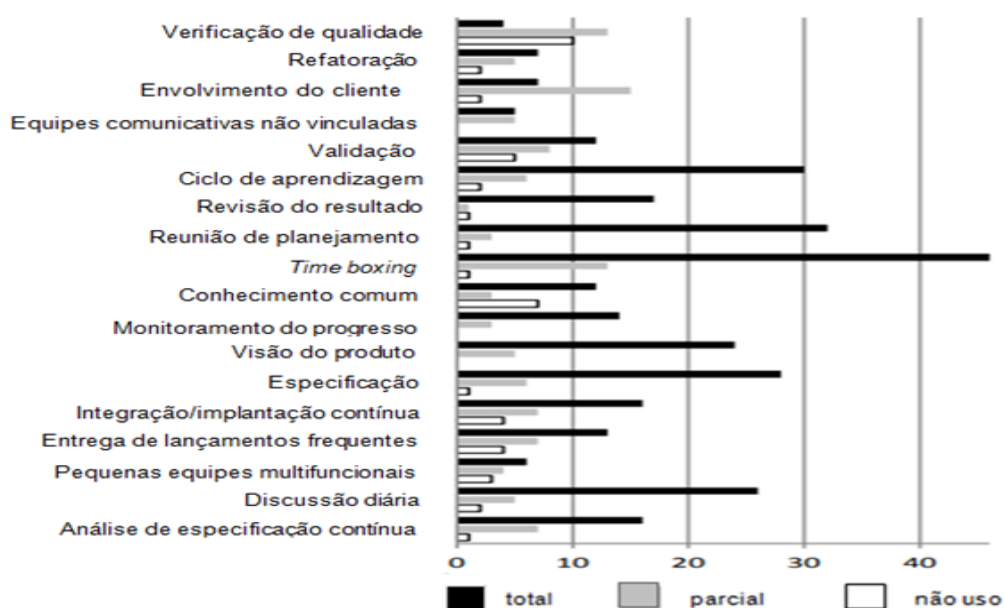


Figura 4 - Uso geral de práticas ágeis - Fonte: Diebold et al. (2014)

Em todos os 68 projetos, todas as 18 práticas ágeis foram usadas. As barras pretas no gráfico mostram que existem seis práticas mais usadas: time boxing, reunião de planejamento, loop de aprendizado, especificação evolutiva e hierárquica, práticas adaptadas diariamente discussão(ões) e visão do produto. Para práticas adaptadas ou parcialmente usadas, há apenas três que ultrapassam 10 usos: envolvimento do cliente, time boxing e verificação de qualidade (DIEBOLD et al.,2014).

Durante a pesquisa alguns artigos relataram sobre o uso de processos ágeis e suas combinações: combinação de *Scrum* e XP, a combinação de *Scrum*, XP e *Lean*, *Waterfall* enriquecido com *Scrum*, o processo com o menor número de práticas mencionadas é o *Kanban*, em alguns casos sendo combinado com *Scrum*. Além do *Agile Software Development* combinado com *Lean* ou com *Scrum*. Sem dúvida o *Scrum* foi o processo mais mencionado nos artigos analisados (CICO et al, 2020) e DIEBOLD et al,2014).

Tavares et al (2019), realizaram estudo sobre métodos ágeis com o objetivo de propor uma lista de práticas de gerenciamento de riscos para projetos ágeis, visando aumentar as chances de sucesso. O gerenciamento de riscos contribui para o sucesso dos projetos de *software*, mas os métodos ágeis de desenvolvimento de *software* não oferecem atividades específicas para gerenciar os riscos. O estudo apresenta as práticas de gerenciamento de riscos para reuniões diárias, incremento, protótipo, *backlog* do produto e planejamento do *Sprint* como as mais importantes para a efetividade do gerenciamento de riscos.

Foram usados componentes e subcomponentes do *Scrum* para classificar as práticas de gerenciamento de risco. À medida que identificaram práticas de outros métodos ágeis, criaram novos componentes e subcomponentes para classificá-los. Usaram cinco componentes e 48 subcomponentes para classificar as 127 práticas de gerenciamento de risco. Cinco componentes usados como categorias foram: artefatos, recursos, eventos, papéis e técnicas e métodos.

Segundo pesquisa realizada, os artefatos são o componente mais importante para a realização de uma gestão de riscos eficaz em projetos ágeis, seguidos de eventos, funcionalidades, papéis e técnicas e métodos (TAVARES et al, 2019).

Tabela III - Justificativas para os componentes mais importantes

Componente	Justificativas	Respondentes
Artefatos	Responsável pelo registro de riscos, sua exposição e planos de resposta.	80%
	Existem artefatos críticos para a eficácia do gerenciamento de riscos, como o repositório de riscos.	30%
	As estimativas de escopo, cronograma e custo são artefatos e os principais fatores de risco.	20%
Eventos	Implementar a identificação, análise, resposta e monitoramento de riscos.	90%
	A criação de artefatos ocorre durante os eventos.	50%
	Enfatize a comunicação e possibilite discutir e antecipar os possíveis riscos da iteração e do projeto.	10%

Fonte: Tavares et al. (2019)

Na tabela IV os subcomponentes classificados como totalmente recomendados e o número de práticas de gerenciamento de risco dentro deles. São os subcomponentes mais importantes para o gerenciamento de riscos em projetos ágeis e, portanto, devemos implementar suas práticas para aumentar as chances de sucesso de nossos projetos.

Tabela IV - Subcomponentes totalmente recomendados

Ordem	Subcomponente	Resultado	Práticas
1	Encontro diário	7.40%	6
2	Incremento	5.93%	6
3	Protótipo	5.68%	2
4	Product backlog (Lista de pendências do produto)	5.54%	7
5	Planejamento de Sprint	5.32%	2
6	Backlog da Sprint	4.69%	4
7	Refinamento do backlog do produto	4.32%	2
8	Reunião semanal de risco	4.28%	1
9	Especificação técnica	4.23%	4

Fonte: Tavares et al. (2019)

Na tabela V os subcomponentes fortemente recomendados, na opinião dos respondentes, esses subcomponentes e suas práticas têm forte importância para a eficácia da gestão de riscos.

Tabela V - Subcomponentes fortemente recomendados

Ordem	Subcomponente	Resultado	Práticas
10	Sprint	3.53%	11
11	Repositório de riscos	3.25%	3
12	Comunicação e colaboração contínuas	3.23%	6
13	Revisão do <i>sprint</i>	2.95%	2
14	Alinhamento de negócios	2.92%	1
15	Plano de contingência	2.82%	1
16	Proprietário do produto	2.57%	12
17	Retrospectiva do <i>sprint</i>	2.18%	1
18	Atualização do projeto	2.14%	2
19	Equipe de desenvolvimento	2.11%	25

Fonte: Tavares et al. (2019)

Na tabela VI os subcomponentes moderadamente recomendados, este nível de recomendação tem o menor número de práticas de gestão de risco.

Tabela VI - Subcomponentes moderadamente recomendados

Ordem	Subcomponente	Resultado	Práticas
20	Biblioteca de códigos de <i>software</i>	1,69%	2
21	<i>Wiki</i> - ferramenta de edição colaborativa	1,41%	1
22	Análise de viabilidade de custos	1.38%	1
23	Análise qualitativa e quantitativa	1.36%	1
24	Equipe multifuncional	1.32%	2

25	Equipe auto-organizada	1.27%	1
26	Integração contínua	1.10%	1
27	TDD - <i>Test Driven Development</i>	1.08%	1
29	Alocação de recursos	1.05%	1
30	SWOT - <i>Strengths</i> (Forças), <i>Weaknesses</i> (Fraquezas), <i>Opportunities</i> (Oportunidades) e <i>Threats</i> (Ameaças)	1.04%	1
31	Abordagem orientada a riscos	1.01%	1

TDD - (Desenvolvimento Orientado por Testes) - Fonte: Tavares et al. (2019)

Na tabela VII os subcomponentes com o menor nível de recomendação.

Tabela VII - Subcomponentes recomendados

Ordem	Subcomponente	Resultado	Práticas
32	Transparência do artefato	0.92%	4
33	Proprietário da arquitetura	0.88%	4
34	<i>Kanban</i>	0.84%	1
35	Adaptação	0.84%	1
36	Analista de negócios	0.78%	2
37	Gestor de projeto	0.74%	2
38	<i>Scrum master</i>	0.73%	15
39	Coordenador Técnico	0.72%	2
40	Líder da equipe	0.71%	1
41	Abordagem orientada a valor	0.66%	1
42	Programação em pares	0.60%	1
43	<i>Business visionary</i> (Visionário de negócios)	0.45%	4
44	Marco do projeto	0.43%	3
45	Escala de tempo	0.23%	1
46	DSDM - <i>Dynamic Systems Development Method</i>	0.22%	1
47	MoSCow - <i>must have, should have, could have e wouldn't have</i>	0.19%	1

48	Testes	0.16%	3
----	--------	-------	---

DSDM - (Metodologia de Desenvolvimento de Sistemas Dinâmicos)

MoSCow - (tenho que fazer, devo fazer, poderia fazer, não será feito (por enquanto))

Fonte: Tavares et al. (2019)

A reunião diária é o subcomponente mais importante para a gestão de riscos em projetos ágeis, durante um planejamento de *Sprint* é possível identificar riscos, analisar exposições a riscos e desenvolver planos de resposta (TAVARES et al, 2019).

O incremento é o segundo subcomponente mais recomendado para gerenciar riscos. Este artefato reduz a probabilidade de falhas e aumenta as chances de sucesso do projeto, fornecendo um rápido *feedback* do cliente (TAVARES et al, 2019).

Incremento e protótipo são os artefatos mais importantes para gerenciar riscos em métodos ágeis. Um incremento fornece *feedback* do cliente, permitindo identificar novos riscos. E, um protótipo também fornece *feedback* do cliente relacionado aos requisitos do produto e permite identificar e reduzir a exposição a riscos(TAVARES et al, 2019).

O proprietário do produto e a equipe de desenvolvimento desempenham os papéis mais importantes para gerenciar riscos em métodos ágeis. O proprietário do produto tem um papel importante para gerenciar os riscos do negócio, enquanto a equipe de desenvolvimento é importante para gerenciar os riscos técnicos (TAVARES et al, 2019).

A comunicação e colaboração contínuas é a característica mais importante, pois a redução de artefatos nos métodos ágeis aumenta a necessidade de comunicação e colaboração entre os membros da equipe. Além disso, o alinhamento do negócio é a segunda característica mais importante, pois a falta dele é um dos principais motivos do insucesso do projeto(TAVARES et al, 2019).

Técnicas e métodos são recomendados ou moderadamente recomendados. A análise de viabilidade de custos é a mais importante e apresenta a seguinte prática de gestão de risco: focar apenas em recursos que possam ser justificados pelos custos. A análise qualitativa e quantitativa é a segunda técnica e método mais importante para a gestão de riscos(TAVARES et al, 2019).

Os resultados deste estudo contribuem para a melhoria da gestão de riscos em projetos ágeis e, conseqüentemente, aumentam suas chances de sucesso (TAVARES et al, 2019).

East et al (2019), da Empresa Accenture, relatam que os desenvolvedores de produtos enfrentam desafios quando tentam adotar métodos ágeis, pois os métodos estão bem estabelecidos para equipes de *software*, mas a aplicação é muito mais rara no nível do produto de *hardware*. A utilização de abordagem híbrida no projeto geral de desenvolvimento do produto executando um processo em cascata, limitando o *Agile* aos desenvolvedores de *software*, poderá sobrecarregar a equipe de *software* com grandes fases de planejamento e requisitos iniciais, negando completamente os principais benefícios do método Agile.

Para que equipes multidisciplinares (desenvolvimento de produtos e engenheiros de *hardware*), utilizem a abordagem *Agile* e capturem seus benefícios, a Accenture sugere quatro passos principais (EAST et al, 2019):

- **Especificação de Requisitos do Produto**

Abordagem por declarações leves expressas da perspectiva do usuário final. A verificação é feita com o *feedback* capturado de demonstrações de produtos frequentes.

Utilizar “história do usuário” que as equipes de *software Agile* empregam. Mantendo a rastreabilidade com o *design* cuidadoso do processo e da ferramenta de colaboração. Além disso, os benefícios de uma fase de especificação mais curta superam em muito a falta de fidelidade que alguns desenvolvedores de produtos temem.

- **Priorize o conjunto mínimo de características diferenciadas do produto e demonstre cedo e frequentemente**

Priorizar um pequeno subconjunto de funcionalidades do produto significa que as equipes podem demonstrar rapidamente a funcionalidade, buscar *feedback* dos usuários e validar suposições iniciais de preferências do usuário e viabilidade técnica. Essa abordagem é a base para metodologias de desenvolvimento de *software* como *Scrum*. Usamos a ferramenta Núcleo de Produtos para estabelecer prioridades de produtos, começamos estabelecendo as necessidades do usuário que seu produto deve atender, em ordem de importância. A priorização é fundamental – sem ela, pequenas diferenças na incorporação do produto previsto são descobertas profundamente no desenvolvimento quando a mudança é dispendiosa.

- **Adote um ritmo de desenvolvimento fixo para alinhar o planejamento e integração em todas as equipes**

Um dos aspectos mais conhecidos dos métodos ágeis é o *timebox* de desenvolvimento fixo ou '*sprint*'. As equipes de *software* trabalham em incrementos de duração fixa que variam de uma a quatro semanas. Durante esse período, os desenvolvedores planejarão, desenvolverão, testarão e fornecerão um recurso (ou história) de *software* completo. O trabalho de incremento curto facilita o *feedback* rápido porque as equipes ágeis sempre procurarão entregar algo demonstrável aos usuários finais, em vez de componentes de *software* parcialmente completos.

Uma solução comprovada em campo para equipes de engenharia multidisciplinares é a abordagem dupla de *timebox* defendida pelo *Scaled Agile Framework* (SAFe™). SAFe descreve um *sprint* curto ideal para equipes de *software*, mas alinha o planejamento e a integração com as equipes de *hardware* por meio de um *timebox* mais longo conhecido como *Program Increment*.

- **Invista em modelos de produtos complexos para acelerar a descoberta de problemas e dissociar as equipes**

Para dar o próximo passo e tornar-se verdadeiramente *Agile*, os desenvolvedores de produtos precisam dissociar as equipes de *hardware* e *software* para que possam executar simultaneamente, mas tenham certeza de que suas soluções integradas atenderão às especificações de nível de sistema.

A Empresa Accenture acredita que com processo cuidadoso e *design* organizacional, os métodos ágeis podem funcionar para equipes de desenvolvimento de produtos de *hardware* (EAST et al., 2019).

4.1 Análise do Contexto Atual da Engenharia Software na Indústria

Nos estudos analisados referentes a engenharia de *software* no contexto da indústria, identificamos que o uso do desenvolvimento ágil de *software* tem sido cada vez mais adotado nas últimas duas décadas. Porém, grande parte dos estudos mostram que o uso de métodos ágeis não tem sido praticado conforme definido na literatura, são adaptados pelas empresas as quais usam apenas partes deles (chamadas práticas ágeis). Há práticas que são utilizadas com mais frequência, sendo que a aplicação de diferentes práticas pode ser influenciada pelo domínio e o processo.

Os métodos mais usados para desenvolvimento ágil são *Scrum* e *Extreme Programming* (XP). A maioria das empresas adapta esses métodos devido a vários

problemas, como complexidade encontrada durante a introdução ou mudança para o desenvolvimento ágil. Na maioria das vezes, eles omitem partes específicas do método ágil original, as alteram ou as substituem por aspectos tradicionais.

Durante a pesquisa alguns artigos relataram sobre o uso de processos ágeis e suas combinações: combinação de *Scrum* e *XP*, a combinação de *Scrum*, *XP* e *Lean*, *Waterfall* enriquecido com *Scrum*, o processo com o menor número de práticas mencionadas é o *Kanban*, em alguns casos sendo combinado com *Scrum*. Além do *Agile Software Development* combinado com *Lean* ou com *Scrum*. Sem dúvida o *Scrum* foi o processo mais mencionado nos artigos analisados.

Durante a pesquisa foi encontrado estudo sobre métodos ágeis com o objetivo de propor uma lista de práticas de gerenciamento de riscos para projetos ágeis. Uma vez que os métodos ágeis de desenvolvimento de *software* não oferecem atividades específicas para gerenciar os riscos. O estudo apresenta as práticas de gerenciamento de riscos para reuniões diárias, incremento, protótipo, *backlog* do produto e planejamento do *Sprint* como as mais importantes para a efetividade do gerenciamento de riscos. Foram usados como categorias: artefatos, recursos, eventos, papéis e técnicas e métodos.

Neste estudo os artefatos foram considerados o componente mais importante para a realização de uma gestão de riscos eficaz em projetos ágeis, seguidos de eventos, funcionalidades, papéis e técnicas e métodos. Além de uma lista de subcomponentes para o gerenciamento de riscos seguindo a seguinte classificação: totalmente recomendados, fortemente recomendados e recomendados. Com essa classificação percebe-se que apesar de não haver atividades específicas para gerenciar os riscos, pode-se fazê-lo durante os ciclos do processo.

5. ANÁLISE DO PPC (Projeto Pedagógico de Curso) e PLANOS DE ENSINO DAS DISCIPLINAS DO EIXO DA ENGENHARIA DE SOFTWARE DO IFRS - RESTINGA

Conforme análise realizada no Projeto Pedagógico do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (PPC) do Instituto Federal do Rio Grande do Sul (IFRS) - Campus Restinga (IFRS, 2017), foram encontradas

as informações que seguem: o curso nomeado Análise e Desenvolvimento de Sistemas (ADS), que têm um período de seis semestres, tem como objetivo preparar os estudantes tecnicamente buscando estimular a compreensão global da situação e despertar um espírito crítico, proporcionando aprendizados relacionados à capacidade deste em analisar, planejar e implementar projetos na área de Tecnologia da Informação. Além de valorizar as competências escrita e humanística, o raciocínio lógico e matemático, a capacidade de liderar e trabalhar em equipe, dando ênfase para os aspectos ligados aos processos de análise e desenvolvimento de sistemas, trazendo a ética como base para a atividade profissional e a necessidade de constante aperfeiçoamento pessoal e profissional.

Dentre as capacidades previstas a serem desenvolvidas ao longo do período do curso, destaca-se a de trabalhar em equipe entendendo as relações humanas nas organizações e identificar oportunidades de negócio, criar e gerenciar empreendimentos relacionados a sistemas de informação e tecnologia da informação.

Durante toda a pesquisa realizada esta foi uma das habilidades mais mencionadas nos artigos analisados como necessária aos graduados.

Além de prever atividades práticas durante todo o curso, através dos componentes curriculares voltados ao Desenvolvimento de Sistemas, permite que os demais componentes curriculares dos grupos Hardware e Redes de Computadores, Análise de Sistemas e Desenvolvimento de Sistemas possam prever atividades práticas. Possibilitando integração teórica e prática e a interdisciplinaridade, assegurando a sólida formação dos estudantes para a prática profissional.

Ao analisar os Planos de Ensino e as ementas do eixo da ES pode-se afirmar que o conteúdo previsto no PPC do curso, contém grande parte das práticas e tendências encontradas nos estudos. Como o uso do Scrum, XP e Lean Software Development, testes de software, prototipagem, usabilidade, gestão de qualidade, bem como o uso de práticas de software, conforme a necessidade da indústria, entre outras.

Com relação aos Planos de Ensino, em sua maioria não há a previsão de prática de laboratório simulando situações reais da indústria e a interdisciplinaridade, com exceção da disciplina Desenvolvimento de Sistemas II onde a turma é dividida em grupos para o desenvolvimento de um sistema fazendo

uso das práticas, técnicas e ferramentas aprendidas durante o curso. O projeto envolve a disciplina de Desenvolvimento de Sistemas II e ES.

No Plano de Ensino da disciplina de Teste de *Software* prevê projeto envolvendo o desenvolvimento de um software, bem como a prática de testes com o objetivo de aplicar os conceitos aprendidos.

Os demais Planos de Estudo preveem teoria, aplicação de questionários, videoaulas, fóruns, trabalhos, provas, participação em palestras da Semana Acadêmica, sem mencionar projeto envolvendo a interdisciplinaridade ou de problematização de situações de rotina da indústria.

Para uma melhor preparação dos graduandos do curso de ADS do IFRS - Campus Restinga, sugere-se a adoção de projetos que envolvam disciplinas afins para que o conteúdo possa ser trabalhado em um maior período de tempo, além da simulação do ambiente da indústria, para que quando os alunos ingressarem no mercado de trabalho possam estar mais preparados. Saliencia-se também, que essas práticas farão com que os alunos desenvolvam habilidades de convivência em equipe, comunicação, gerenciamento, entre outras.

6. CONCLUSÃO

Foi realizada uma revisão bibliográfica na qual iniciou-se os estudos a procura de artigos científicos que trouxessem explicitamente o que estava sendo trabalhado tanto na academia quanto na indústria, porém foram encontrados poucos artigos nesse sentido. E em sua maioria relatando essa mesma dificuldade.

Nos estudos analisados com ênfase na educação, foram relatados o uso de laboratórios simulando ambientes reais utilizando *Scrum* e XP, segundo os quais resultaram em uma experiência positiva fazendo com que os graduandos desenvolvessem as habilidades para atender as necessidades da indústria.

Os artigos analisados na área da indústria trouxeram afirmações quanto ao uso do desenvolvimento ágil de *software*, em especial o *Scrum*, porém há relatos de adaptações conforme a necessidade/realidade do eixo em que está sendo usado, as chamadas práticas ágeis. Essas práticas podem contar com o uso parcial de um método, bem como a combinação deles. Também encontrou-se estudos do uso de métodos ágeis no gerenciamento de riscos, bem como a aplicação do *Scrum* na produção de produtos de *hardware*.

A análise realizada no PPC e planos de ensino das disciplinas de ES do curso superior em ADS do IFRS - Campus Restinga constatou-se que as ementas das disciplinas preveem o ensino da maioria das práticas apontadas nos artigos, todavia no que diz respeito às estratégias de ensino, há poucos relatos de projetos envolvendo as disciplinas entre si. Já os planos de ensino poderiam conter maior detalhamento do conteúdo trabalhado e prever projetos que envolvam a interdisciplinaridade e/ou o conteúdo já trabalhado anteriormente, mesmo que não seja atual disciplina, pois assim os alunos poderão fixar melhor o aprendizado.

Conclui-se que há a necessidade de mais pesquisas no sentido de manter a academia mais atualizada e próxima das necessidades e práticas da indústria. Além da academia buscar incentivo/parceria para projetos de laboratórios e na falta dessa possibilidade tentar implantar algo dentro da sua realidade.

Como trabalhos futuros pode-se sugerir a realização de uma *survey* envolvendo academia e indústria.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMI, N. E. A. M., Rahman, N. A., Purusothaman, D., & Sulaiman, S. (2011). Software engineering education: The gap between industry's requirements and graduates' readiness. 2011 IEEE Symposium on Computers & Informatics. doi:10.1109/isci.2011.5958974

BASS, M. (2016). Software Engineering Education in the New World: What Needs to Change? 2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET). doi:10.1109/cseet.2016.34

CICO, O., Jaccheri, L., Nguyen-Duc, A., & Zhang, H. (2020). Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. Journal of Systems and Software, 110736. doi:10.1016/j.jss.2020.110736

DAGNINO, A. (2014). Increasing the effectiveness of teaching software engineering: A University and industry partnership. 2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T). doi:10.1109/cseet.2014.6816781

DE O. MELO, C., Santos, V., Katayama, E., Corbucci, H., Prikladnicki, R., Goldman, A., & Kon, F. (2013). The evolution of agile software development in Brazil. Journal of the Brazilian Computer Society, 19(4), 523–552. doi:10.1007/s13173-013-0114-x

DESENVOLVIMENTO AGIL, Scrum - Desenvolvimento Ágil do Brasil, 2013/2014. Disponível em: <<http://www.desenvolvimentoagil.com.br/scrum/>>. Acesso em: 08 de fevereiro de 2022.

DIEBOLD, P., & Dahlem, M. (2014). Práticas ágeis na prática. Anais da 18ª Conferência Internacional de Avaliação e Avaliação em Engenharia de Software - EASE '14. doi:10.1145/2601248.2601254

EAST, A., Roberts, S., Myers, S., Beaumont, M., & Clavin, K. (2019). Beyond Software: Applying Agile Principles to Connected Product Development,

https://www.accenture.com/_acnmedia/pdf-103/accenture-beyond-software-applying-agile-principles-to-connected-product-development-pov-new.pdf.

EQUIPE RUNRUN.IT, Metodologia lean: descubra como enxugar os excessos da gestão, 2013. Disponível em: <<https://blog.runrun.it/metodologia-lean/>>. Acesso em: 08 de fevereiro de 2022.

FIA - Fundação Instituto de Administração, Scrum: o que é e como aplicar a metodologia ágil para gestão?, 2020. Disponível em: <<https://fia.com.br/blog/scrum/>>. Acesso em: 11 de fevereiro de 2022.

FONTES, Aléxia, Lean: o que é?, 2018. Disponível em: <<https://voitto.com.br/blog/artigo/lean-o-que-e>>. Acesso em: 08 de fevereiro de 2022.

G. TARAN and M. Rosso-Llopart, "Software Engineering Education in Russia: A Comparative Study of People, Process and Technology a Four Year Perspective", 20th Conference on Software Engineering Education & Training (CSEET), IEEE Computer Society, July 2007, pp. 19-28.

GAROUSI, V., Petersen, K., Ozkan, B., 2016 . Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. Inf. Softw. Technol. 79 , 106 - 127.

GDF - Escola de Governo - Governo do Distrito Federal, Metodologias Ágeis/Scrum/Lean, 2021. Disponível em: <<https://egov.df.gov.br/metodologias-ageis-scrum-lean/>>. Acesso em: 08 de fevereiro de 2022.

IFRS - Instituto Federal do Rio Grande do Sul - Campus Restinga, Projeto Pedagógico do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (PPC), 2017. Disponível em: <<https://restinga.ifrs.edu.br/site/midias/arquiv>

os/20171122104136422projeto_pedagogico_curso_ads_resolucao_101_consup.pdf>. Acesso em: 12 de fevereiro de 2022.

MARQUES, M., & Robledo, J. (2018). What Software Engineering “Best Practices” are we Teaching Students - a Systematic Literature Review. 2018 IEEE Frontiers in Education Conference (FIE). doi:10.1109/fie.2018.8658576

MEDEIROS, Higor. Introdução ao Extreme Programming (XP), 2013. Disponível em: <<https://www.devmedia.com.br/introducao-ao-extreme-programming-xp/29249>>. Acesso em: 08 de fevereiro de 2022.

O'LEARY, C., Lawless, D., Gordon, D., Li Haifeng, & Bechkoum, K. (nd). Desenvolvendo um Currículo de Engenharia de Software para a Indústria de Software Emergente na China. 19ª Conferência sobre Educação e Treinamento em Engenharia de Software (CSEET'06). doi:10.1109/cseet.2006.16

R. B. Vaughn and J. Carver, “Position Paper: The Importance of Experience with Industry in Software Engineering Education”, Proceeding of the 19th Conference on Software Engineering Education and Training Workshops 06 (CSEETW), IEEE Computer Society, April 2006, p. 19.

REVISTA JAVA MAGAZINE 81, Lean Software Development - Java Magazine 81, 2010, <https://www.devmedia.com.br/lean-software-development-java-magazine-81/17442>

SCRUM, O QUE É SCRUM?, Scrum.org, 2022. Disponível em: <<https://www.scrum.org/resources/what-is-scrum>>. Acesso em: 07 de fevereiro de 2022.

TAVARES, B. G., da Silva, C. E. S., & de Souza, A. D. (2019). Practices to Improve Risk Management in Agile Projects. International Journal of Software Engineering and Knowledge Engineering, 29(03), 381–399. doi:10.1142/s0218194019500165

TOMÁS, Mário Rui (2009): Métodos ágeis: características, pontos fortes e fracos e possibilidades de aplicação, Research Centre on Enterprise and Work Innovation, IET Working Papers Series, 09/2009, 19 pp.

TOTVS, Metodologia ágil: o que é e como implementar, 2021. Disponível em: <<https://www.totvs.com/blog/negocios/metodologia-agil/>>. Acesso em: 11 de fevereiro de 2022.

YAMAGUTI, M. H., de Oliveira, F. M., Trindade, C. A. W., & Dutra, A. C. S. (2017). AGES. Proceedings of the 31st Brazilian Symposium on Software Engineering - SBES'17. doi:10.1145/3131151.3131187

ANEXOS



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

PLANO DE ENSINO

1. IDENTIFICAÇÃO

Curso: Superior de Tecnologia Análise e Desenvolvimento de Sistemas
Semestre: 3º
Turno: noite
Ano letivo: 2021-2
Componente curricular: Interface Humano-Computador
Carga horária total do componente curricular (igual da do PPC): 66h
Ementa: Fundamentos de modelagem de interação. Avaliação de sistemas interativos. Projeto de interface. Modelagem de interfaces. Conceitos de multimídia e sistemas multimídia. Noções de ambientes de realidade virtual. Usabilidade. Acessibilidade.
Professor(a) responsável pela oferta do componente: Diego Moreira da Rosa
Email para contato do(a) professor(a): diego.rosa@restinga.ifrs.edu.br

2. METODOLOGIA, RECURSOS DIDÁTICOS E TECNOLÓGICOS

Serão realizadas atividades remotas na modalidade síncrona (através do Google Meet) e assíncrona (através do Moodle). Todos os conteúdos e atividades das aulas síncronas serão também disponibilizados na página do curso no Moodle. Para o melhor aproveitamento do curso, recomenda-se que os estudantes tenham à disposição: computador pessoal (notebook ou desktop) com sistema operacional atualizado e conexão com a Internet.

As aulas serão organizadas de forma semanal e todo o material estará no Moodle. Os encontros síncronos através do Google Meet ocorrerão nos seguintes horários: terça-feira 19:30-21:00 (aula síncrona) e terça-feira 21:00-22:00 (horário de atendimento).



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

3. CONTEÚDO E CRONOGRAMA

PERÍODO	OBJETIVOS DE APRENDIZAGEM	CONTEÚDO(S)	ATIVIDADE(S)	CH
Semana 0: 06/10/21 a 10/10/21	Compreender os processos do Campus no semestre corrente	Introdução aos processos do Campus	Atividades de acolhimento e ambientação dos estudantes	---
Feriado 12/10 (Aparecida)	Feriado	Feriado	Feriado	---
Semana 1: 18/10 a 24/10 (terça-feira)	Conhecer aspectos do mercado de trabalho	O mercado de trabalho de informática	Participação nas palestras da Semana Acadêmica dos Cursos de Informática	3,3h
Semana 2: 18/10 a 24/10 (sáb. letivo)	Compreender as bases da interação humano-computador	Introdução à Interação Humano-Computador	Atividade de leitura e lista de exercícios no Moodle	3,3h
Semana 3: 25/10 a 31/10	Compreender as abordagens teóricas de IHC	Abordagens teóricas de IHC	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h
Feriado: 01/11 a 07/11	Feriado (Finados)	Feriado (Finados)	Feriado (Finados)	---
Semana 4: 08/11 a 14/11	Compreender as abordagens teóricas de IHC	Abordagens teóricas de IHC e Teoria das Cores	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h
Semana 5: 15/11 a 21/11	Planejar o processo de design de IHC	Processos de design, análise competitiva	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h
Semana 6: 22/11 a 28/11	Compreender o processo de coleta de requisitos de IHC	Requisitos de IHC	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h
Semana 7: 29/11 a 05/12	Compreender e construir diferentes modelos de IHC	Modelos de IHC, organização do espaço de problema, análise de tarefas	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h
Semana 8: 06/12 a 12/12	Projetar interfaces de software aplicando os conceitos de IHC	Design de IHC, níveis de abstração	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h
Semana 9: 13/12 a 19/12	Aplicar princípios e diretrizes no processo de design	Princípios e diretrizes de IHC	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

	de IHC			
Semana 10: 20/12 a 26/12	Compreender o conceito de acessibilidade e sua aplicação em sistemas interativos	Acessibilidade no design de interação	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h
Semana 11: 27/12 a 02/01/2022	Planejar a avaliação de interfaces de software	Planejamento da avaliação de IHC, métodos de avaliação	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h
Semana 12: 03/01 a 09/01/2022 (terça-feira)	Avaliar interfaces de software utilizando métodos de inspeção	Métodos de inspeção, Avaliação Heurística	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h
Semana 13: 03/01 a 09/01/2022 (sáb. letivo)	Avaliar interfaces de software utilizando métodos de observação	Métodos de observação, testes com usuários, Teste de Usabilidade	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h
Semana 14: 10/01 a 16/01	Avaliar interfaces de software utilizando métodos de observação	Métodos de observação, testes com usuários, Teste de Usabilidade	Atividades assíncronas (exercícios)	3,3h
Semana 15: 17/01 a 23/01	Avaliar interfaces de software	Outros métodos de avaliação	Aula expositiva por vídeo, atividades de leitura no Moodle, lista de exercícios	3,3h
Semana 16: 24/01 a 30/01	Adquirir uma visão ampla do design de interação	Visão geral do design de interação	Tempo para trabalho nos projetos	3,3h
Semana 17: 31/01 a 06/02 (terça-feira)	Adquirir uma visão ampla do design de interação	Visão geral do design de interação	Apresentação e discussão dos trabalhos finais	3,3h
Semana 18: 31/01 a 06/02 (sáb. letivo)	Adquirir uma visão ampla do design de interação	Visão geral do design de interação	Apresentação e discussão dos trabalhos finais	3,3h
Semana 19: 07/02 a 13/02	Adquirir uma visão ampla do design de interação	Visão geral do design de interação	Apresentação e discussão dos trabalhos finais	3,3h
Semana 20: 14/02 a 15/02	Recuperar conteúdos	Todos os conteúdos do semestre	Exame	3,3h3



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

4. AVALIAÇÃO

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
Princípios de IHC e processo de design de IHC	Exercícios disponibilizados no Moodle semanalmente	Exercícios serão avaliados de acordo com o conteúdo e qualidade das respostas	Todos os exercícios devem ser entregues até as 23:59 do dia 28/01
Planejamento de projetos de IHC	Trabalho a ser realizado individualmente sobre o planejamento do projeto de uma interface	Trabalho será avaliado com base na qualidade do planejamento apresentado	O trabalho deve ser entregue até as 23:59 do dia 19/12
Elaboração de projetos de IHC	Trabalho a ser realizado individualmente sobre a construção de um projeto de interface	Trabalho será avaliado com base na qualidade do projeto apresentado	O trabalho deve ser entregue até as 23:59 do dia 23/01

Cada um dos instrumentos de avaliação será avaliado com uma nota de 0 (zero) a 10: Exercícios (EX), Trabalho 1 (T1) e Trabalho 2 (T2). A avaliação final dos estudantes será dada pela média aritmética simples entre os três instrumentos. O estudante que atingir a média final (MF) igual ou superior a 7,00 terá atingido os objetivos propostos:

$$MF = (TA + T1 + T2) / 3 \geq 7,00$$

ATIVIDADES DE RECUPERAÇÃO PARALELA

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
Processo de design e avaliação de IHC	Avaliação formativa realizada durante os horários de atendimento	Participação do estudante nos horários de atendimento e evolução observada pelo docente	Os atendimentos ocorrerão semanalmente
Processo de design e avaliação de IHC	Avaliação somativa realizada no último dia de aula (prova no Moodle)	A prova será avaliada conforme o conteúdo e qualidade das respostas.	A prova será realizada na última semana de aula

5. ATENDIMENTO AOS ESTUDANTES

O atendimento aos estudantes será realizado de forma assíncrona através de email e fórum no Moodle e também de forma síncrona em horário pré-definido utilizando a ferramenta Google Meet. O horário de atendimento síncrono será nas terças-feiras, das 21:00 às 22:00.

6. BIBLIOGRAFIAS



MINISTÉRIO DA EDUCAÇÃO
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Restinga

6.1 Bibliografia Básica

BARBOSA, S.D.J.; SILVA, B.S. **Interação Humano-Computador**. Editora Campus- Elsevier, 2010.

PREECE, J.; ROGERS, Y.; SHARP, H. **Design de Interação: Além da Interação Humano-Computador**. Porto Alegre: Bookman, 2005.

SOUZA, C. S. de. **The Semiotic Engineering of Human-Computer Interaction**. Cambridge, MA. The MIT Press, 2005.

6.2 Bibliografia Complementar

NIELSEN, J. **Usability Engineering**. Boston, MA: Academic Press, 1993.

NORMAN, D.A.; DRAPER, S.W. **User Centered System Design**. Hillsdale, NJ. Lawrence Erlbaum and Associates, 2005.

OLIVEIRA NETTO, A. A. **IHC – Interface Humano Computador – Modelagem e Gerência de Interfaces com o Usuário**. Editora Visual Books, 2004.

SHNEIDERMAN, B. **Designing the user interface: strategies for effective human-computer interaction**. 5a. ed. Reading, MA: Addison-Wesley, 2010.

WINOGRAD, T. **Bringing Design to Software**. Reading, MA: Addison-Wesley, 1996

7. OBSERVAÇÕES

Alterações a este plano de ensino podem ocorrer de acordo com a necessidade e combinação entre o professor e a turma. Quaisquer alterações serão notificadas através do Moodle e email.

Porto Alegre, 27 de outubro de 2021.

Digitally signed by
Diego Moreira da Rosa
Date: 2021-11-08 16:
58:42

Prof. MSc. Diego Moreira da Rosa



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

PLANO DE ENSINO

1. IDENTIFICAÇÃO

Curso: Técnico em Análise e Desenvolvimento de Software
Semestre: 3 semestre
Turno: Noite
Ano letivo: 2021/2
Componente curricular: Engenharia de Software
Carga horária total do componente curricular (igual da do PPC): 66h
Carga horária <u>presencial</u> do componente curricular (igual a do PPC): 66h
Carga horária <u>a distância</u> do componente curricular (igual a do PPC): 0
Ementa: Contextualização da Engenharia de Software. Fundamentação dos Princípios da Engenharia de Software. Conceituação de Produto e Processo de Software. Comparação entre os Paradigmas de Desenvolvimento Software. Caracterização do Projeto de Software. Introdução a Gerenciamento de Projetos. Introdução a Qualidade de Software.
Professor responsável pela oferta do componente: Eliana Pereira
Email para contato do(a) professor(a): eliana.pereira@restinga.ifrs.edu.br

2. METODOLOGIA, RECURSOS DIDÁTICOS E TECNOLÓGICOS

Será disponibilizada todas as semanas uma vídeo aula e um material complementar que poderá ser um podcast, um artigo, vídeo, capítulo de livro entre outros. Para todas as semanas estão programadas atividades para assimilação do conteúdo trabalhado. Estas atividades são questionários, fóruns e trabalhos práticos. Todas as atividades contam como participação na disciplina e algumas delas serão avaliativas. Todas as semanas haverá pelo menos um encontro virtual com duração de uma a duas horas.



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

3. CONTEÚDO E CRONOGRAMA

PERÍODO	OBJETIVOS DE APRENDIZAGEM	CONTEÚDO (s)	ATIVIDADE	CARGA HORÁRIA (hora-relógio)
Semana 1	Semana de Acolhida dos estudantes	Semana de Acolhida dos estudantes	Palestras e Cursos	3,3
Semana 2	Entendimento de todo histórico e importância da engenharia de software. Entendimento dos problemas envolvidos no desenvolvimento de software	- Introdução a Disciplina e Introdução a Eng. de Software	Assistir as vídeo aulas e Interagir respondendo perguntas no fórum do moodle.	3,3
Semana 3	Semana Acadêmica ADS	Semana Acadêmica ADS	Palestras	3,3



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 4	<p>Entendimento do conceito de ciclo de vida de software</p> <p>Entendimento dos principais ciclos de vida para o desenvolvimento de software</p>	<p>Ciclo de Vidas e Processos de Desenvolvimento de Software</p>	<p>Assistir as vídeo aulas, leitura complementar de artigo, questionário sobre ciclo de vida com entrega pelo moodle</p> <p>Aula síncrona</p>	3,3
Semana 5	<p>Entendimento sobre a processos tradicionais de desenvolvimento de software</p>	<p>Processos de Desenvolvimento de Software</p> <p>Processo Unificado</p>	<p>Assistir as vídeo aulas, estudar o material complementar, Interagir respondendo perguntas num fórum do moodle.</p> <p>Aula síncrona</p>	3,3
Semana 6	<p>Entendimento sobre a transição do uso dos métodos/processos tradicionais para os métodos ágeis</p> <p>Conhecer o manifesto ágil</p>	<p>Introdução aos Métodos Ágeis</p> <p>- Manifesto Ágil</p>	<p>Assistir as vídeo aulas, estudar o material complementar, Interagir respondendo perguntas no fórum do moodle.</p> <p>Aula síncrona</p>	3,3
Semana 7	<p>Conhecer a metodologia Extremme programming (ciclo de vida iterativo e incremental, papéis, práticas, princípios)</p>	<p>- Extremme Programming</p>	<p>Assistir as vídeo aulas, estudar o material complementar, Interagir respondendo perguntas no fórum do moodle e questionário com entrega pelo moodle.</p>	3,3



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

			Aula síncrona	
Semana 8	Conhecer a metodologia SCRUM (ciclo de vida iterativo e incremental, papéis, product backlog, sprint, retrospectiva, revisão, ferramentas)	Scrum – parte 1	Assistir as vídeo aulas, estudar o material complementar e questionário com entrega pelo moodle. Aula síncrona	3,3
Semana 9	Conhecer a metodologia SCRUM (ciclo de vida iterativo e incremental, papéis, product backlog, sprint, retrospectiva, revisão, ferramentas)	Scrum – parte 2	Assistir as vídeo aulas, estudar o material complementar e trabalho sobre SCRUM. Aula síncrona	3,3
Semana 10	Conhecer a metodologia Kanban e o uso do quadro de tarefas nos métodos ágeis	Kanban quadro de tarefas	Assistir as vídeo aulas, exercícios TRabalho sobre scrum Aula síncrona	3,3
Semana 11	Conhecer a filosofia Lean e a metodologia Lean Startup	Lean Lean Startup	Assistir as vídeo aulas, estudar o material complementar e questionário com entrega pelo moodle. Aula síncrona	3,3



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 12	Entendimento do conceito de requisitos, req funcionais, req não funcionais, fases da engenharia de requisitos	Introdução a requisitos	Assistir as vídeo aulas, estudar o material complementar, Interagir respondendo perguntas num fórum do moodle. Aula síncrona	3,3
Semana 13	Entendimento sobre a fase de coleta de requisitos Técnicas de coleta/elicitación de requisitos	Elicitación de requisitos	Assistir as vídeo aulas, estudar o material complementar e trabalho sobre Técnicas de coleta/elicitación de requisitos. Aula síncrona	3,3
Semana 14	Entendimento sobre a fase de modelagem e documentação de requisitos	Modelagem e documentação de requisitos	Assistir as vídeo aulas, estudar o material complementar e trabalho sobre modelagem de requisitos. Aula síncrona	3,3
Semana 15	Entendimento sobre escrita de requisitos (casos de uso e história de usuário)	Modelagem e documentação de requisitos	Assistir as vídeo aulas, estudar o material complementar e trabalho sobre modelagem de requisitos. Aula síncrona	3,3



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 16	Entendimento sobre escrita de requisitos (casos de uso e história de usuário) - continuação	Modelagem e documentação de requisitos	Assistir as vídeo aulas e exercícios Aula síncrona	3,3
Semana 17	Design thinking (técnicas para descoberta de requisitos)	design thinking - duplo diamante - técnicas: persona, mapa do usuário, storytelling	Assistir as vídeo aulas e exercí Aula síncrona	3,3
Semana 18	Design thinking (técnicas para descoberta de requisitos) - continuação	design thinking - duplo diamante - técnicas: persona, mapa do usuário, storytelling	Assistir as vídeo aulas e exercí Aula síncrona	3,3
Semana 19	Entendimento dos conceitos de verificação e validação	Verificação e Validação	Assistir as vídeo aulas Instrumento avaliação da disciplina Aula síncrona	3,3



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 20	Exame	todo conteúdo da disciplina	avaliação final	3,3
Total de carga horária (horas-relógio)				66

4. AVALIAÇÃO

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
Scrum	Trabalho sobre ferramentas SCRUM	identificar o grau de entendimento dos alunos quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	03/11
Elicitação requisitos	Trabalho sobre técnicas de elicitação	identificar o grau de entendimento dos alunos quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	29/12
Modelagem requisitos	Trabalho sobre modelagem de	identificar o grau de entendimento dos alunos	19/01



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

	requisitos (história de usuário)	quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	
--	----------------------------------	---	--

ATIVIDADES DE RECUPERAÇÃO PARALELA

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
Scrum	Trabalho sobre ferramentas SCRUM	identificar o grau de entendimento dos alunos quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	15/11
Elicitação requisitos	Trabalho sobre técnicas de elicitação	identificar o grau de entendimento dos alunos quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	10/01
Modelagem requisitos	Trabalho sobre modelagem de requisitos (história de usuário)	identificar o grau de entendimento dos alunos quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	03/02

5. ATENDIMENTO AOS ESTUDANTES

O atendimento aos estudantes será realizado através de e-mails e um fórum de dúvidas criado no moodle. Também haverá atendimento através de encontro virtual a ser realizado conforme demanda dos alunos

6. BIBLIOGRAFIAS

6.1 Bibliografia Básica

PFLIEGER, S. L., Engenharia de Software: Teoria e Prática. 2ª edição. São Paulo: Pearson



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Education. 2004.

PRESSMAN, R.S. Engenharia de Software. 6ª edição Rio de Janeiro: McGraw-Hill, 2011.

SOMMERVILLE, I. Engenharia de Software. 8ª edição. São Paulo: Addison Wesley, 2007.

6.2 Bibliografia Complementar

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML Guia do usuário. 2ª edição. Rio de Janeiro: Campus, 2012.

LARMAN, Craig. Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3ª edição. Bookman, 2007.

MAGELA, Rogerio. Engenharia de Software Aplicada: Princípios (volume 1). Alta Books. 2006.

MAGELA, Rogerio. Engenharia de Software Aplicada: Fundamentos (volume 2). Alta Books. Paula Filho, W. Engenharia de Software: Fundamentos, Métodos e Padrões – 3ª edição - LTC - 2009.

7 . OBSERVAÇÕES

Porto Alegre, 19 de outubro de 2021.

Eliana Beatriz
 Pereira

Assinado digitalmente por Eliana Beatriz Pereira
 DN: C=BR, OU=Coordenação do Curso Técnico
 em Informática, O=IFRS - Campus Restinga,
 CN=Eliana Beatriz Pereira,
 E=eliana.pereira@restinga.ifrs.edu.br
 Razão: Eu sou o autor deste documento
 Localização: rolia@1
 Data: 2021-10-21 15:22:10
 Foxit Reader Versão: 9.3.0

Eliana Beatriz Pereira



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

PLANO DE ENSINO

1. IDENTIFICAÇÃO

Curso: Tecnólogo em Análise e Desenvolvimento de Software
Semestre: 4 semestre
Turno: Manhã
Ano letivo: 2021/2
Componente curricular: Análise e Projeto de Software
Carga horária total do componente curricular (igual da do PPC): 66h
Carga horária <u>presencial</u> do componente curricular (igual a do PPC): 66h
Carga horária <u>a distância</u> do componente curricular (igual a do PPC): 0
Ementa: Introdução à análise e projeto de sistemas. Caracterização e aplicação de metodologias e ferramentas de modelagem de sistemas orientados a objetos. Apresentação e aplicação de uma metodologia desenvolvimento de sistemas de software.
Professor responsável pela oferta do componente: Eliana Pereira
Email para contato do(a) professor(a): eliana.pereira@restinga.ifrs.edu.br

2. METODOLOGIA, RECURSOS DIDÁTICOS E TECNOLÓGICOS

Será disponibilizada todas as semanas uma vídeo aula e um material complementar que poderá ser um podcast, um artigo, vídeo, capítulo de livro entre outros. Para todas as semanas estão programadas atividades para assimilação do conteúdo trabalhado. Estas atividades são questionários, fóruns e trabalhos práticos. Todas as atividades contam como participação na disciplina e algumas delas serão avaliativas. Todas as semanas haverá pelo menos um encontro virtual com duração de uma a duas horas.



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

3. CONTEÚDO E CRONOGRAMA

PERÍODO	OBJETIVOS DE APRENDIZAGEM	CONTEÚDO (s)	ATIVIDADE	CARGA HORÁRIA (hora-relógio)
Semana 1	Semana de Acolhida dos estudantes	Semana de Acolhida dos estudantes	Palestras e Cursos	3,3
Semana 2	Conceituação de modelagem e UML	- Introdução a Disciplina e Introdução a Modelagem UML	Assistir as vídeo aulas Aula Síncrona	3,3
Semana 3	Semana Acadêmica ADS	Semana Acadêmica ADS	Palestras	3,3



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 4	Diagrama de Caso de Uso	Atores caso de uso extends includes herança	Assistir as vídeo aulas, exercícios Aula síncrona	3,3
Semana 5	Diagrama de Atividades	Elementos diagrama de atividades	Assistir as vídeo aulas, exercícios Aula síncrona	3,3
Semana 6	Diagrama de Classe - parte 1	- objeto - classe - método - atributo	Assistir as vídeo aulas, exercícios Aula síncrona	3,3
Semana 7	Diagrama de Classe - parte 2	- relacionamentos diagrama de classes - modelo conceitual	Assistir as vídeo aulas, exercícios Aula síncrona trabalho da disciplina 1	3,3



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 8	Diagrama de Classe - parte 3	- modelo de análise e projeto revisão	Assistir as vídeo aulas, exercícios trabalho da disciplina 1 Aula síncrona	3,3
Semana 9	Diagrama de objeto	Elementos diagrama de objeto	Assistir as vídeo aulas, exercícios trabalho da disciplina 1 Aula síncrona	3,3
Semana 10	Diagrama de interação	- sequência - interação geral - comunicação - elementos diagrama sequencia	Assistir as vídeo aulas, exercícios Aula síncrona	3,3
Semana 11	Diagrama de sequência	Elementos diagrama de sequencia	Assistir as vídeo aulas, exercícios Aula síncrona	3,3



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 12	Semana de trabalho e revisão	revisão	Assistir as vídeo aulas, exercícios Aula síncrona	3,3
Semana 13	Diagrama de máquina de estado	Elementos diagrama de máquina de estados	Assistir as vídeo aulas, trabalho da disciplina - parte 2 Aula síncrona	3,3
Semana 14	Diagrama de Pacotes	Elementos diagrama de pacotes	Assistir as vídeo aulas trabalho da disciplina - parte 2 Aula síncrona	3,3
Semana 15	Diagrama de Componentes	Elementos diagrama de componentes	Assistir as vídeo aulas Aula síncrona	3,3



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 16	Diagrama de Implantação	de Elementos diagrama de implantação	Assistir as vídeo aulas e exercícios Aula síncrona	3,3
Semana 17	Revisão final diagramas	- revisão	Assistir as vídeo aulas e exercícios Aula síncrona	3,3
Semana 18	Revisão final diagramas	- revisão	Assistir as vídeo aulas e exercícios trabalho da disciplina - parte 2 Aula síncrona	3,3
Semana 19	Avaliação disciplina	instrumento para avaliar a disciplina	Aula síncrona	3,3



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 20	Exame	todo conteúdo da disciplina	avaliação final	3,3
Total de carga horária (horas-relógio)				66

4. AVALIAÇÃO

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
Diagrama de caso de uso diagrama de atividades Diagrama de Classe	TRabalho para modelar um sistema	identificar o grau de entendimento dos alunos quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	15/12
Diagrama sequencia Diagrama Máquina de Estados	TRabalho para modelar um sistema	identificar o grau de entendimento dos alunos quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	10/01
Diagrama implantação	TRabalho para	identificar o grau de entendimento dos alunos	25/01



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

	modelar um sistema	quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	
--	--------------------	---	--

ATIVIDADES DE RECUPERAÇÃO PARALELA

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
Diagrama de caso de uso diagrama de atividades Diagrama de Classe	TRabalho para modelar um sistema	identificar o grau de entendimento dos alunos quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	26/12
Diagrama sequencia Diagrama Máquina de Estados	TRabalho para modelar um sistema	identificar o grau de entendimento dos alunos quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	20/01
Diagrama implantação	TRabalho para modelar um sistema	identificar o grau de entendimento dos alunos quantos as etapas de conhecer, desenvolver e aplicar o conteúdo proposto. nota de 0 a 10.	05/02

5. ATENDIMENTO AOS ESTUDANTES

O atendimento aos estudantes será realizado através de e-mails e um fórum de dúvidas criado no moodle. Também haverá atendimento através de encontro virtual a ser realizado conforme demanda dos alunos

6. BIBLIOGRAFIAS

6.1 Bibliografia Básica

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: guia do usuário. Rio de



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Janeiro: Elsevier, 2012.

LARMAN. Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientados a Objetos. 3a. ed. Bookman, 2007.

PRESSMAN, R.S. Engenharia de Software. 7a ed. Rio de Janeiro: McGraw-Hill, 2011.

6.2 Bibliografia Complementar

BEZERRA, E.. Princípios de Análise e Projetos de Sistemas com UML. Campus, 2007.

BRAUDE, E. Projeto de Software: da Programação à Arquitetura. Porto Alegre: Bookman, 2005.

FOWLER, M. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. 3ª ed. Porto Alegre: Bookman, 2005.

SOMMERVILLE, Ian. Engenharia de software. São Paulo : Pearson Prentice Hall, 2011.

WAZLAWICK, Raul Sidnei. Análise e projeto de sistemas de informação orientados a objetos. 2. ed. rev. atual. Rio de Janeiro: Elsevier, 2011. 330 p.

7 . OBSERVAÇÕES

Porto Alegre, 19 de outubro de 2021.

**Eliana Beatriz
 Pereira**

Assinado digitalmente por Eliana Beatriz Pereira
 DN: C=BR, OU=Coordenação do Curso Técnico em
 Informática, O=IFRS - Campus Restinga, CN=Eliana
 Beatriz Pereira, E=eliana.pereira@restinga.ifrs.edu.br
 Razão: Eu sou o autor deste documento
 Localização: rolrat@1
 Data: 2021-10-21 15:22:36
 Perfil Reader Versão: 9.3.0

Eliana Beatriz Pereira



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

PLANO DE ENSINO

1. IDENTIFICAÇÃO

Curso: Análise e Desenvolvimento de Sistemas
Semestre: 4º semestre
Turno: Manhã
Ano letivo: 2021/2
Componente curricular: Estratégia por Projetos
Carga horária total do componente curricular (igual da do PPC): 66 horas
Carga horária presencial do componente curricular (igual a do PPC): 66 horas
Carga horária a distância do componente curricular (igual a do PPC): -
Ementa: Balanced Scorecard (BSC). As perspectivas do BSC. Objetivos estratégicos. Mapas Estratégicos. Indicadores de Ocorrência e de Tendência. Portfólio, Programas e Projetos. Gerenciamento de Projetos (Guia PMBOK). Estudo dos grupos de processos. Estudo das nove áreas de conhecimento.
Professor(a) responsável pela oferta do componente: Nilson Varella Rübenich
E-mail para contato do(a) professor(a): nilson.rubenich@restinga.ifrs.edu.br

2. METODOLOGIA, RECURSOS DIDÁTICOS E TECNOLÓGICOS

a) Atividades assíncronas disponibilizadas por meio do AVA Moodle contemplando recursos digitais (textos teóricos e práticos, vídeos, resolução de exercícios, estudos de casos, avaliações, fóruns temáticos etc.);

b) Atividades síncronas quinzenais disponibilizadas por meio de ferramentas de encontros virtuais ex.: "google meet" etc. (Atividade Síncrona - AS).



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

3. CONTEÚDO E CRONOGRAMA

PERÍODO	OBJETIVOS DE APRENDIZAGEM	CONTEÚDO (s)	ATIVIDADE(S)	CARGA HORÁRIA (hora-relógio)
06/out a 12/out	Introdução: compreender a relação entre Gestão Estratégica e Gestão de Projetos	* Artigo: "Por que é possível utilizar os projetos para tornar sua gestão mais estratégica?"	Leitura de materiais, vídeos e atividade de aprendizagem	3,5 horas
13/out a 19/out	Compreender conceitos fundamentais de estratégia	* Conceitos fundamentais de estratégia	Leitura de materiais, vídeos e atividade de aprendizagem Encontro síncrono	3,5 horas
20/out a 26/out	Conhecer conceitos e estrutura do Balanced Scorecard (BSC)	* BSC: conceitos e histórico * BSC: Estrutura	Leitura de materiais, vídeos e atividade de aprendizagem	3,5 horas
27/out a 02/nov	Compreender Mapas Estratégicos	* Mapas Estratégicos * Objetivos e iniciativas estratégicas * Indicadores estratégicos	Leitura de materiais, vídeos e atividade de aprendizagem Encontro síncrono	3,5 horas
03/nova 09/nov	Verificação de Aprendizagem	Avaliação 1 - Questionário	Avaliação 1 - Questionário	3,5 horas
10/nova 16/nov	Conhecer conceitos iniciais da Gestão de Projetos e o PMBOK	* GP - Fundamentos * Importância da gestão de projetos no contexto atual	Leitura de materiais, vídeos e atividade de aprendizagem Encontro síncrono	3,5 horas
17/nova 23/nov	Conhecer os conceitos de PMO, PMI, PMP e GP	* PMI/PMBOK * GP, PMP	Leitura de materiais, vídeos e atividade de aprendizagem	3,5 horas
24/nova 30/nov	Compreender ciclos de vida de projetos e	* Ciclo de vida do projeto	Leitura de materiais, vídeos e atividade de	3,5 horas



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

	os processos de gerenciamento de projetos	* Processos de gerenciamento de projetos Avaliação 2 - Questionário	aprendizagem Encontro síncrono Avaliação 2 - Questionário	
01/dez a 07/dez	Compreender o que é o escopo, os riscos e as premissas de um projeto	* Gerenciamento do Escopo * Riscos e Premissas * Termo de Abertura de Projeto	Leitura de materiais, vídeos e atividade de aprendizagem	3,5 horas
08/dez a 14/dez	Conhecer as áreas do Conhecimento do PMBOK	* Gerenciamento de Riscos * Gerenciamento das Partes Interessadas * Gerenciamento da integração do Projeto	Leitura de materiais, vídeos e atividade de aprendizagem Encontro síncrono	3,5 horas
15/dez a 21/dez	Conhecer as áreas do Conhecimento do PMBOK	* Gerenciamento do Tempo / Cronograma Trabalho 1 - definições iniciais do projeto	Leitura de materiais, vídeos e atividade de aprendizagem Trabalho 1 - definições iniciais do projeto	3,5 horas
22/dez a 28/dez	Conhecer as áreas do Conhecimento do PMBOK	* Gerenciamento dos Custos / Orçamento	Leitura de materiais, vídeos e atividade de aprendizagem	3,5 horas
29/dez a 04/jan	Conhecer as áreas do Conhecimento do PMBOK	* Gerenciamento de RH e Comunicação	Leitura de materiais, vídeos e atividade de aprendizagem	3,5 horas
05/jan a 11/jan	Conhecer as áreas do Conhecimento do PMBOK	* Gerenciamento da Qualidade e Aquisições	Leitura de materiais, vídeos e atividade de aprendizagem Encontro síncrono	3,5 horas
12/jan a 18/jan	Compreender o que são métodos ágeis	* Movimento ágil * SCRUM	Leitura de materiais, vídeos e atividade de aprendizagem	3,5 horas



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

19/jan a 25/jan	Conhecer softwares/APPs de gestão de projetos	* Softwares e APPs de Gestão de Projetos	Leitura de materiais, vídeos e atividade de aprendizagem Encontro síncrono	3,5 horas
26/jan a 01/fev	Integrar os conhecimentos de Gestão de Projetos	Apresentação do Trabalho final - Plano de Projeto	Apresentação de Trabalhos via Google Meet Encontro síncrono	3,5 horas
02/fev a 08/fev	Integrar os conhecimentos de Gestão de Projetos	Entrega do Trabalho final - Plano de Projeto	Entrega do plano de projeto via Moodle	3,5 horas
09/fev	Auto avaliação e Exame	Auto avaliação e Exame	Auto avaliação Avaliação do componente Exame	3 horas
Total de carga horária (horas-relógio)				66 horas

4. AVALIAÇÃO

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
Atividades de aprendizagem	Questionários, fóruns e tarefas semanais	Participação – 10% da nota final	Todas as semanas, exceto as que possuem outra avaliação
Avaliação 1	Questionário no moodle	Acertos – 20% da nota final	Semana 5 – 03 a 09/11/2021
Avaliação 2	Questionário no moodle	Acertos – 25% da nota final	Semana 8 – 24 a 30/11/2021
Avaliação 3	Trabalho parcial - definições iniciais do projeto	Coerência com o conteúdo e com as instruções do trabalho; criatividade para o tema do projeto – 10% da nota final	Semana 11 – 15 a 21/12/2021
Avaliação 4	Trabalho final - Plano de Projeto	Coerência com o conteúdo e com as instruções do trabalho;	Semana 18 – 02 a 08/02/2022



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

		organização do trabalho escrito; criatividade para as soluções de gestão de projeto apresentadas – 25% da nota final	
Avaliação 5	Apresentação do Trabalho final - Plano de Projeto	Coerência com o conteúdo e com as instruções do trabalho; domínio do trabalho; desenvoltura e clareza na apresentação; materiais gráficos de suporte para apresentação – 10% da nota final	Semana 17 - 10%

ATIVIDADES DE RECUPERAÇÃO PARALELA

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
A combinar caso necessárias.	Exercícios e atendimento individual.	Conforme o conteúdo.	A combinar

5. ATENDIMENTO AOS ESTUDANTES

O atendimento a(o) estudante se dará por meio de e-mail ou da ferramenta de encontro virtual *google meet* em horário previamente agendado entre estudante e professor, por e-mail. O atendimento ocorrerá preferencialmente nas quartas-feiras, entre 11h e 12h.

6. BIBLIOGRAFIAS

6.1 Bibliografia Básica

KAPLAN, R. S.; NORTON, D. P. A estratégia em ação: Balanced Scorecard. 7 ed. Rio de Janeiro: Campus, 1997.

KAPLAN, R. S.; NORTON, D. P. Mapas estratégicos – Balanced Scorecard: convertendo ativos intangíveis em resultados tangíveis. 6 ed. Rio de Janeiro: Elsevier, 2004.

PROJECT MANAGEMENT INSTITUTE. Um guia do conhecimento em gerenciamento de projetos – guia PMBOK. 4. ed. 2009.

6.2 Bibliografia Complementar

KAPLAN, R. S.; NORTON, D. P. Organização orientada para a estratégia: como as empresas que adotam o Balanced Scorecard prosperam no novo ambiente de negócios.



MINISTÉRIO DA EDUCAÇÃO
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Restinga

Rio de Janeiro: Campus, 2000.

KAPLAN, R. S.; NORTON, D. P. Alinhamento: utilizando o Balanced Scorecard para criar sinergias corporativas. Rio de Janeiro: Elsevier, 2006.

KERZNER, Harrold. Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 8a edi..o, John Wiley & Sons Inc, 2003.

LUECKE, R. Estratégia. Harvard Business Essentials. 1 ed. Record, 2009.

VARGAS, R. V. Manual Prático do Plano de Projeto – Utilizando o PMBOK Guide. 3a ed. São Paulo: Brasport, 2007.

7 . OBSERVAÇÕES

O plano apresentado será avaliado e reavaliado ao longo do desenvolvimento das atividades, podendo sofrer alterações que se julgarem necessárias.

Porto Alegre, 06 de outubro de 2021.

Nilson Varella
Rübenich

Assinado de forma digital
por Nilson Varella Rübenich
Dados: 2021.10.06 10:55:26
-03'00'

Prof. Nilson Varella Rübenich



MINISTÉRIO DA EDUCAÇÃO
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Restinga

PLANO DE ENSINO

1. IDENTIFICAÇÃO

Curso: Técnico em Análise e Desenvolvimento de Software
Semestre: 5 semestre
Turno: Noite
Ano letivo: 2021/2
Componente curricular: Desenvolvimento de Sistemas 2
Carga horária total do componente curricular (igual da do PPC): 83h
Carga horária <u>presencial</u> do componente curricular (igual a do PPC): 66h
Carga horária <u>a distância</u> do componente curricular (igual a do PPC): 17h
Ementa: Gerenciamento de configuração de software. Gerenciamento de Mudanças. Controle de versão. Desenvolvimento de software utilizando uma ferramenta de gerenciamento de configuração. Análise e Modelagem de um Sistema de Informação: Levantamento de Requisitos; Casos de Uso; Modelagem dos Objetos; Modelagem dos Dados; Casos de Testes; Definição de Arquitetura. Desenvolvimento de um Sistema de Informação: Utilização de Linguagens de Programação Orientadas a Objetos; Ambiente de Desenvolvimento; Versionamento; Utilização de Padrões de Projeto e Frameworks; Utilização de Banco de Dados – Mapeamento Objeto-Relacional; Reutilização de Componentes.
Professor responsável pela oferta do componente: Eliana Pereira e Eduarda Monteiro
Email para contato do(a) professor(a): eliana.pereira@restinga.ifrs.edu.br eduarda.monteiro@restinga.ifrs.edu.br

2. METODOLOGIA, RECURSOS DIDÁTICOS E TECNOLÓGICOS

Esta disciplina terá encontros síncronos todas as semanas e algumas vídeo aulas e materiais serão disponibilizados no moodle. Trata-se de uma disciplina prática onde se aplica a metodologia de aprendizagem por projeto. A turma será dividida em grupos e



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

deverá desenvolver um sistema usando metodologias, ferramentas e linguagens aprendidas no curso de ADS nos semestres anteriores.

3. CONTEÚDO E CRONOGRAMA

PERÍODO	OBJETIVOS DE APRENDIZAGEM	CONTEÚDO (s)	ATIVIDADE	CARGA HORÁRIA (hora-relógio)
Semana 1	Semana de Acolhida dos estudantes	Semana de Acolhida dos estudantes	Palestras e Cursos	4,15
Semana 2	Apresentar formato da disciplina	- Introdução a Disciplina e apresentação geral de funcionamento projeto Apresentação Programa	Assistir as vídeo aulas Aula síncrona material de revisão sobre scrum	4,15



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 3	Semana Acadêmica ADS	Semana Acadêmica ADS Formação das equipes de desenvolvimento	Palestras equipes postadas no moodle	4,15
Semana 4	Desenvolvimento do Projeto prático de software	Entrega dos temas dos projetos apresentação template product backlog	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 5	Desenvolvimento do Projeto prático de software	início sprint 0 (ideação e design)	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 6	Desenvolvimento do Projeto prático de software	sprint 0 (ideação e design)	Aula síncrona atividades de desenvolvimento do projeto	4,15



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 7	Desenvolvimento do Projeto prático de software	entrega sprint 0 (ideação e design) revisão da sprint retrospectiva da sprint	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 8	Desenvolvimento do Projeto prático de software	planejamento sprint 1 início sprint 1	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 9	Desenvolvimento do Projeto prático de software	sprint 1	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 10	Desenvolvimento do Projeto prático de software	sprint 1	Aula síncrona atividades de desenvolvimento do projeto	4,15



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 11	Desenvolvimento do Projeto prático de software	sprint 1	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 12	Desenvolvimento do Projeto prático de software	sprint 1	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 13	Desenvolvimento do Projeto prático de software	entrega sprint 1 revisão da sprint retrospectiva da sprint planejamento sprint 2 início sprint 2	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 14	Desenvolvimento do Projeto prático de software	sprint 2	Aula síncrona atividades de desenvolvimento do projeto	4,15



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 15	Desenvolvimento do Projeto prático de software	sprint 2	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 16	Desenvolvimento do Projeto prático de software	sprint 2	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 17	Desenvolvimento do Projeto prático de software	entrega sprint 2 revisão da sprint retrospectiva da sprint planejamento sprint 3 início sprint 3	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 18	Desenvolvimento do Projeto prático de software	sprint 3	Aula síncrona atividades de desenvolvimento do projeto	4,15



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

Semana 19	Desenvolvimento do Projeto prático de software	entrega final projeto revisão e retrospectiva sprint e projeto.	Aula síncrona atividades de desenvolvimento do projeto	4,15
Semana 20	Exame	todo conteúdo da disciplina	avaliação final	4,15
Total de carga horária (horas-relógio)				83

4. AVALIAÇÃO

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
sprint 0	TRabalho referente a etapa 1 do projeto (sprint 0)	Nota em grupo de 0 a 10 Nota individual de 0 a 10 Todos serão avaliados em grupo e de forma individual qto as entregas	15/11



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Restinga

		do projeto, participação em equipe, comunicação.	
sprint 1	TRabalho referente a etapa 2 do projeto (sprint 1)	<p>Nota em grupo de 0 a 10</p> <p>Nota individual de 0 a 10</p> <p>Todos serão avaliados em grupo e de forma individual qto as entregas do projeto, participação em equipe, comunicação.</p>	20/12
sprint 2	TRabalho referente a etapa 2 do projeto (sprint 2)	<p>Nota em grupo de 0 a 10</p> <p>Nota individual de 0 a 10</p> <p>Todos serão avaliados em grupo e de forma individual qto as entregas do projeto, participação em equipe, comunicação.</p>	17/01
sprint 3	TRabalho referente a etapa 3 do projeto (sprint 3)	<p>Nota em grupo de 0 a 10</p> <p>Nota individual de 0 a 10</p> <p>Todos serão avaliados em grupo e de forma individual qto as entregas do projeto, participação em equipe, comunicação.</p>	04/02

ATIVIDADES DE RECUPERAÇÃO PARALELA

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
sprint 0	TRabalho referente a etapa 1 do projeto (sprint 0)	<p>Nota em grupo de 0 a 10</p> <p>Nota individual de 0 a 10</p> <p>Todos serão avaliados em grupo e de forma</p>	20/11



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Restinga

		individual qto as entregas do projeto, participação em equipe, comunicação.	
sprint 1	TRabalho referente a etapa 2 do projeto (sprint 1)	Nota em grupo de 0 a 10 Nota individual de 0 a 10 Todos serão avaliados em grupo e de forma individual qto as entregas do projeto, participação em equipe, comunicação.	25/12
sprint 2	TRabalho referente a etapa 2 do projeto (sprint 2)	Nota em grupo de 0 a 10 Nota individual de 0 a 10 Todos serão avaliados em grupo e de forma individual qto as entregas do projeto, participação em equipe, comunicação.	22/01
sprint 3	TRabalho referente a etapa 3 do projeto (sprint 3)	Nota em grupo de 0 a 10 Nota individual de 0 a 10 Todos serão avaliados em grupo e de forma individual qto as entregas do projeto, participação em equipe, comunicação.	08/02

5. ATENDIMENTO AOS ESTUDANTES

O atendimento aos estudantes será realizado através de e-mails e um fórum de dúvidas criado no moodle. Também haverá atendimento através de encontro virtual a ser realizado conforme demanda dos alunos

6. BIBLIOGRAFIAS

6.1 Bibliografia Básica



MINISTÉRIO DA EDUCAÇÃO
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Restinga

DEITEL, Paul J.; DEITEL, Harvey M. Ajax, Rich Internet Applications e desenvolvimento Web para programadores. Pearson Education, 2009.

DEITEL, H. M.; DEITEL, P. J.; NIETO, T. R. Internet and World Wide Web: Como Programar. Bookman, 2003.

DEITEL, Paul J. C++ Como Programar. 5ª edição. Prentice Hall Brasil, 2006.

6.2 Bibliografia Complementar

GONÇALVES, Edson. Desenvolvendo Aplicações Web com JSP, SERVELTS, JAVA SERVER FACES, HIBERNATE, EJB 3 PERSISTENCE E AJAX. 1ª edição, Editora Ciência Moderna, 2007.

BAUER, Christian; KING, Gavin. Java Persistence com Hibernate. Ciência Moderna, 2007.

FOWLER, M. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. 3ª ed. Porto Alegre: Bookman, 2005.

BRAUDE, E. Projeto de Software: da Programação à Arquitetura. Porto Alegre: Bookman, 2005.

MOLINARI, Leonardo. Gerência de Configuração: Técnicas e Práticas no Desenvolvimento de Software. 1 ed. Florianópolis: Visual Books, 2007.

7 . OBSERVAÇÕES

Porto Alegre, 19 de outubro de 2021.

**Eliana Beatriz
Pereira**

Assinado digitalmente por Eliana Beatriz Pereira
DN: C=BR, OU=Coordenação do Curso Técnico em Informática, O=IFRS - Campus Restinga, CN=Eliana Beatriz Pereira,
E=eliana.pereira@restinga.ifrs.edu.br
Razão: Eu sou o autor deste documento
Localização: roliat@1
Data: 2021.10.21 18:07:19
Foxit Reader Versão: 9.3.0

**Eduarda
Monteiro**

Assinado digitalmente por Eduarda Monteiro
DN: C=BR, OU=IFRS - CAMPUS RESTINGA, CN=Eduarda Monteiro,
E=eduarda.monteiro@restinga.ifrs.edu.br
Razão: Eu sou o autor deste documento
Localização: Porto Alegre
Data: 2021.10.21 15:33:41-03'00'
Foxit PDF Reader Versão: 11.0.1

Eduarda Rodrigues Monteiro



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Restinga
 Rua Alberto Hoffmann, 285 – Restinga – Porto Alegre/RS – CEP 91.791-508
 Telefone: (51) 3247.8400 – www.restinga.ifrs.edu.br – E-mail: gabinete@restinga.ifrs.edu.br

PLANO DE ENSINO

Modalidade: () Concomitante () PROEJA () Subsequente (X) Superior	
Curso: Superior de Tecnologia em Análise e Desenvolvimento de Sistemas	
Turma: ADS5	Turno: noite
Disciplina: Teste de Software	
Professor (a): Iuri Albandes Cunha Gomes	Ano e semestre letivos: 2021/2
Carga horária total (semestral): 66 horas	Total de horas-aula: 80 horas-aula
Carga horária presencial: 66 horas	Carga horária a distância: 0
Horário dos períodos letivos: quarta-feira, das 08:20 às 12:00	
Ementa da disciplina:	
Entender os conceitos envolvidos nas técnicas de teste de software, bem como sua aplicação prática.	
Objetivo(s) da disciplina:	
Introdução ao teste de software. Tipos de defeitos. Reportando e rastreando defeitos. Planejamento e Gerenciamento da atividade de teste. Teste funcional. Teste estrutural. Verificação de software com ênfase em normatização e documentação de testes; Teste BlackBox: Control-Flow Testing; Loop Testing; Data-Flow Testing; Transaction-Flow Testing; Domain Testing; Syntax Testing; Finite-State Testing; Statistical Testing e Component Testing. Ferramentas e Automação de teste de software.	
Conteúdos programáticos:	
Aula	Programa de aula
1	Testes caixa branca, Testes caixa preta
2	Testes caixa branca, Testes caixa preta
3	Tabela de decisão
4	Tabela de decisão
5	ferramenta de gestão de testes (testlink)
6	descrição de casos de teste
7	Início de trabalho. Desenvolvimento de software.
8	conclusão e envio do documento de testes (projeto de testes) referentes a P1
9	Avaliação P1
10	Apresentação do trabalho-avaliação P1 da disciplina.
11	Aula destinada para exercicios
12	ferramentas para testes automatizados
13	Trabalho sobre ferramentas para testes automatizados. (apresentação e documento)
14	Trabalho sobre ferramentas para testes automatizados. (apresentação e documento)



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga
 Rua Alberto Hoffmann, 285 – Restinga – Porto Alegre/RS – CEP 91.795-508
 Telefone: (51) 3247.8400 – www.restinga.ifrs.edu.br – E-mail: gabinete@restinga.ifrs.edu.br

15	Ferramenta de testes automatizados (SELENIUM)
16	Ferramenta de testes automatizados (SELENIUM)
17	Aula destinada para trabalho
18	Apresentação de Trabalhos
19	Avaliação P2
20	Avaliação P2 (recuperação)

Metodologia(s) de ensino:

As metodologias adotadas serão aulas expositivas em laboratório de informática utilizando data show e quadro branco intercaladas com atividades práticas de projeto de interfaces e o acompanhamento dos alunos na resolução de exercícios.

Avaliação da aprendizagem:

Critérios:

A avaliação procura identificar o grau de entendimento dos alunos quanto às etapas de conhecer, desenvolver e aplicar o conteúdo proposto, envolvendo os fundamentos necessários para o projeto e avaliação de interfaces.

Instrumentos:

A média semestral será composta das seguintes avaliações:

- Participação em sala de aula e trabalhos realizados ao longo do semestre (T).
- Primeira Avaliação (P1).
- Segunda Avaliação (P2).

Os trabalhos feitos em aula ao longo do trimestre serão avaliados considerando seu conteúdo e data de entrega:

- Trabalho entregue em até 8 dias após a aula: nota máxima 10,0;
- Trabalho entregue em até 15 dias após a aula: nota máxima 8,0;
- Trabalho entregue depois de 15 dias após a aula: nota máxima 6,0.

A média semestral (MS) será calculada pela média aritmética das avaliações conforme a fórmula abaixo:

$$\bullet \text{ MS} = (T + P1 + P2) / 3 \geq 7,0$$

O aluno que tenha frequência mínima de 75% e não atingir MS igual ou superior a 7,0 ao final do período letivo terá direito a exame final (EF) nessa disciplina. O aluno também deve obter MS mínima de 1,8 para poder realizar exame final (EF). A média final (MF) será calculada a partir da nota obtida no exame final (EF) com peso 4 e da nota obtida na média semestral (MS) com peso 6, conforme a equação abaixo:

$$\bullet \text{ MF} = (\text{MS} * 0,6) + (\text{EF} * 0,4) \geq 5,0$$

A aprovação do aluno dar-se-á somente com uma frequência mínima de 75% no período letivo e média semestral (MS) igual ou superior a 7,0 (sete) ou média final (MF) igual ou superior a 5,0 (cinco), quando necessário realizar o exame final.

Cronograma de atividades a distância

Não possui.

Mecanismos de atendimento aos estudantes



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga
 Rua Alberto Hoffmann, 285 – Restinga – Porto Alegre/RS – CEP 91.795-508
 Telefone: (51) 3247.8400 – www.restinga.ifrs.edu.br – E-mail: gabinete@restinga.ifrs.edu.br

O estudantes serão atendidos no horário extra-classe pelo professor. Também será disponibilizada a assistência do Setor de Ensino.

Estudos de Recuperação

Serão oferecidos estudos orientados em horários extra-classe. Será oferecida aula de revisão como preparação para o Exame Final.

Bibliografia básica:

MOLINARI, Leonardo. Testes de Software: Produzindo Sistemas Melhores e Mais Confiáveis. São Paulo: Érica, 2008.

PEZZÈ, Mauro; YOUNG, Michal. Teste e Análise de Software. Porto Alegre: Bookman, 2008.

PRESSMAN, R.S. Engenharia de Software. 5a ed. Rio de Janeiro: McGraw-Hill, 2002.

Bibliografia complementar:

BLACK, Rex. Managing The Testing Process. 3 ed, Wiley, 2009.

DELAMARO, M.E.; MALDONADO, J. C.; JINO, M. Introdução ao Teste de Software. Rio de Janeiro: Campus, 2007.

MOLINARI, Leonardo. Inovação e Automação de Testes de Software. São Paulo: Érika, 2010.

MOLINARI, Leonardo. Gerência de Configuração: Técnicas e Práticas no Desenvolvimento de Software. 1 ed. Florianópolis: Visual Books, 2007.

NEUBERT, Marden. Cvs - Guia De Consulta Rápida. São Paulo: Novatec, 2004.

Horário(s) dos estudos orientados (atendimento):

Quarta-feira das 18:00 às 19:00.

Porto Alegre, 6 de dezembro de 2021.

Juni Albarde

Assinatura do (a) professor (a)



MINISTÉRIO DA EDUCAÇÃO
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Restinga

PLANO DE ENSINO

1. IDENTIFICAÇÃO

Curso: Superior em Análise e Desenvolvimento de Sistemas
Semestre: 6
Turno: manhã
Ano letivo: 2021/2
Componente curricular: Gestão de Processos e Serviços
Carga horária total do componente curricular: 66 h
Carga horária presencial do componente curricular: 66 h
Carga horária à distância do componente curricular:
Ementa: Introdução à análise e projeto de sistemas. Caracterização e aplicação de metodologias e ferramentas de modelagem de sistemas orientados a objetos. Apresentação e aplicação de uma metodologia desenvolvimento de sistemas de software.
Professor responsável pela oferta do componente: Gilberto Pavani
E-mail para contato do professor: gilberto.pavani@restinga.ifrs.edu.br

2. METODOLOGIA, RECURSOS DIDÁTICOS E TECNOLÓGICOS

A disciplina será conduzida através de aulas teóricas e práticas com uso de e-mail, redes sociais e apoio do ambiente educacional Moodle.

3. CONTEÚDO E CRONOGRAMA

PERÍODO	OBJETIVOS DE APRENDIZAGEM	CONTEÚDOS	ATIVIDADE	CARGA HORÁRIA (hora-relógio)
06/10/2021	Compreender e aplicar os conceitos abordados	Introdução à análise de sistemas	Atividades teóricas e práticas mediadas pelo ambiente educacional Moodle	3:20 h
13/10/2021		Introdução ao projeto de sistemas		3:20 h
20/10/2021		Caracterização de metodologias de modelagem de sistemas orientados a objetos		3:20 h



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
 Campus Restinga

27/10/2021		Aplicação de metodologias de modelagem de sistemas orientados a objetos		3:20 h
30/10/2021		Revisão de conteúdos		3:20 h
03/11/2021	Compreender e aplicar os conceitos abordados	Caracterização de ferramentas de modelagem de sistemas orientados a objetos	Atividades teóricas e práticas mediadas pelo ambiente educacional Moodle	3:20 h
10/11/2021		Aplicação de ferramentas de modelagem de sistemas orientados a objetos		3:20 h
17/11/2021		Apresentação de uma metodologia desenvolvimento de sistemas de software		3:20 h
24/11/2021		Aplicação de metodologia desenvolvimento de sistemas de software		3:20 h
24/11/2021		Revisão de conteúdos		3:20 h
01/12/2021		Compreender e aplicar os conceitos abordados		Apresentação Business Process Management
08/12/2021	Aplicação Business Process Management		3:20 h	
15/12/2021	Apresentação Service Oriented Architecture		3:20 h	
22/12/2021	Aplicação Service Oriented Architecture		3:20 h	
29/12/2021	Revisão de conteúdos		3:20 h	



MINISTÉRIO DA EDUCAÇÃO
 Secretaria de Educação Profissional e Tecnológica
 Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Restinga

05/01/2022	Aplicar os conceitos abordados	Atividade prática	Atividades práticas mediadas pelo ambiente educacional Moodle	3:20 h
12/01/2022		Atividade prática		3:20 h
19/01/2022		Revisão de conteúdos		3:20 h
26/01/2022	Avaliação	Prova final		3:20 h
09/02/2022	Avaliação	Exame		3:20 h
Total de carga horária (horas-relógio)				66 h

4. AVALIAÇÃO

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
06/10/2021 a 09/02/2022	Questionários discursivos e de múltipla escolha e trabalhos de pesquisa individuais	Correção, coerência e coesão	06/10/2021 a 09/02/2022

ATIVIDADES DE RECUPERAÇÃO PARALELA

CONTEÚDO	INSTRUMENTO	CRITÉRIOS	DATA/PRAZO
06/10/2021 a 09/02/2022	Questionários discursivos e de múltipla escolha e trabalhos de pesquisa individuais	Correção, coerência e coesão	06/10/2021 a 09/02/2022

5. ATENDIMENTO AOS ESTUDANTES

Atendimento por redes sociais e e-mail conforme calendário e demanda do coletivo de alunos.

6. BIBLIOGRAFIA

6.1 Bibliografia Básica

AALST, Wil van der.: Workflow management: models, methods, and systems. Cambridge: MIT Press, 2002.

BALDAM, R. et al.: Gerenciamento de Processos de Negócios: BPM - Business Process Management. São Paulo: Érica, 2007.

ERL, Thomas. SOA Princípios de Design de Serviços. São Paulo: Pearson Education, 2009.

6.2 Bibliografia Complementar

Business Process Modeling Notation (BPMN). Object Management Group, 2011.



MINISTÉRIO DA EDUCAÇÃO
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Restinga

CRUZ, Tadeu. BPM & BPMS. 2ª edição. Brasport, 2010.

CRUZ, Tadeu. Sistemas, métodos e processos: administrando organizações por meio de processos de negócios. 2a ed. Editora Atlas, 2005.

HANSEN, M. D.: SOA using Java web services. Upper Saddle River: Prentice Hall, 2007.

JOSUTTIS, N.M.: SOA na Prática. São Paulo: Alta Books, 2008.

7. OBSERVAÇÕES

Não há.

Porto Alegre, 19 de outubro de 2021.

A handwritten signature in black ink that reads 'Guilherme João Porani'.

Assinatura do professor