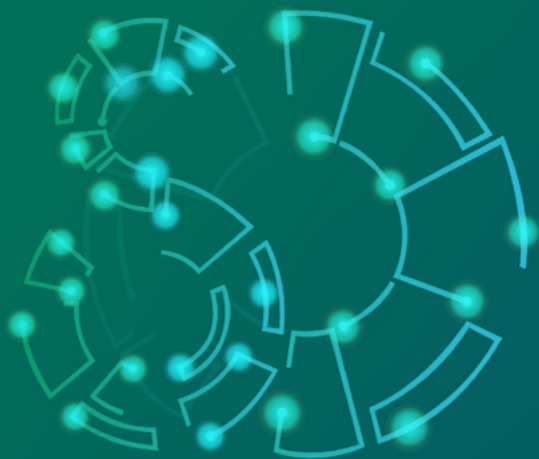




PROGRAMA
EDUCACIONAL
EM **SAÚDE
DIGITAL**
DA UNIVERSIDADE
FEDERAL DE GOIÁS

EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Engenharia de software na saúde digital

Organizadores

Fábio Nogueira de Lucena
Ana Laura de Sene Amâncio Zara
Plínio de Sá Leitão Júnior
Rejane Faria Ribeiro-Rotta
Renata Dutra Braga
Rita Goreti Amaral
Sheila Mara Pedrosa
Silvana de Lima Vieira dos Santos
Taciana Novo Kudo

Cegraf UFG





Universidade Federal de Goiás

Reitora

Angelita Pereira de Lima

Vice-Reitor

Jesiel Freitas Carvalho

Diretora do Cegraf UFG

Maria Lucia Kons

Conselho Editorial da Coleção Programa Educacional em Saúde Digital

Ana Laura de Sene Amâncio Zara (IPTSP / Universidade Federal de Goiás)

Fábio Nogueira de Lucena (INF / Universidade Federal de Goiás)

Gabriella Nunes Neves (CGISD / DATASUS / Secretaria Executiva / Ministério da Saúde)

Merched Cheheb de Oliveira (DATASUS / Secretaria Executiva / Ministério da Saúde)

Juliana Pereira de Souza Zinader (CGISD / DATASUS / Secretaria Executiva / Ministério da Saúde)

Maria Cristina Ferreira de Abreu (CGISD / DATASUS / Secretaria Executiva / Ministério da Saúde)

Rejane Faria Ribeiro-Rotta (FO / Universidade Federal de Goiás)

Renata Dutra Braga (INF / Universidade Federal de Goiás)

Rita Goreti Amaral (FF / Universidade Federal de Goiás)

Sheila Mara Pedrosa (UniEVANGÉLICA)

Silvana de Lima Vieira dos Santos (FEN / Universidade Federal de Goiás)

Taciana Novo Kudo (INF / Universidade Federal de Goiás)

Thais Lucena de Oliveira (CGISD / DATASUS / Secretaria Executiva / Ministério da Saúde)

Equipe de Produção

Amanda Souza Vitor - graduanda (UFG)

Gabriela Martins de Souza - graduanda (UFG)

Iêza Dara Costa Portela - graduada (UFG)

Iuri Vaz Miranda - graduando (UFG)

Jéssica Borges de Carvalho - técnica-administrativa (UFG)

Joyce Beatriz Ferreira da Costa Silva - graduanda (UFG)

Luciana Dantas Soares Alves - analista de TI

Luma Wanderley de Oliveira - doutoranda (UFG)

Patrícia Galúcio Coqueiro Galvão - técnica-administrativa (UFG)

Virgínia de Fernandes Souza - graduanda (UFG)

Suse Barbosa Castilho - mestranda (UFG)

Comissão de Governança da Informação em Saúde (CGIS)

Silvana de Lima Vieira dos Santos

**Centro de Inovação em Gestão da Educação e do Trabalho em Saúde (CIGETS) e
Laboratório de Pesquisa em Empreendedorismo e Inovação (LAPEI)**

Cândido Vieira Borges Júnior

Laboratório de Inovação e Estratégia em Governo (LineGov)

Antônio Isidro da Silva Filho

**Ministério da Saúde / Secretaria Executiva / Departamento de Informática do
Sistema Único de Saúde (DATASUS)**

Merched Cheheb de Oliveira

Esta obra é disponibilizada nos termos da Licença Creative Commons –
Atribuição – Não Comercial – Compartilhamento pela mesma licença
4.0 Internacional. É permitida a reprodução parcial ou total desta obra,
desde que citada a fonte



Engenharia de software na saúde digital

Fábio Nogueira de Lucena

Ana Laura de Sene Amâncio Zara

Plínio de Sá Leitão Júnior

Rejane Faria Ribeiro-Rotta

Renata Dutra Braga

Rita Goreti Amaral

Sheila Mara Pedrosa

Silvana de Lima Vieira dos Santos

Taciana Novo Kudo

(Organizadores)

Cegraf UFG

2022

© Cegraf UFG, 2022

© Fábio Nogueira de Lucena; Ana Laura de Sene Amâncio Zara; Plínio de Sá Leitão Júnior;
Rejane Faria Ribeiro-Rotta; Renata Dutra Braga; Rita Goreti Amaral; Sheila Mara Pedrosa;
Silvana de Lima Vieira dos Santos; Taciana Novo Kudo, 2022

© Universidade Federal de Goiás, 2022

© Ministério da Saúde, 2022

Revisão editorial

Ana Laura Sene Amâncio Zara

Revisão técnica

André Gustavo Souza dos Santos (Ministério da Saúde)
Ana Claudia Sayeg Freire Murahovski (Ministério da Saúde)
Andréia Cristina de Souza Santos (Ministério da Saúde)
Gabriella Nunes Neves (Ministério da Saúde)
Josélio Emar de Araújo Queiroz (Ministério da Saúde)
Juliana Pereira de Souza Zinader (Ministério da Saúde)
Kauara Rodrigues Dias Ferreira (Ministério da Saúde)
Maria Cristina Ferreira de Abreu (Ministério da Saúde)
Patricia dos Santos Irigaray Rodrigues (Ministério da Saúde)
Robson Willian de Melo Matos (Ministério da Saúde)
Thais Lucena de Oliveira (Ministério da Saúde)

Capa

Iuri Vaz Miranda - graduando (UFG)

Editoração Eletrônica

Luma Wanderley de Oliveira - doutoranda (UFG)
Virgínia de Fernandes Souza - graduanda (UFG)

<https://doi.org/10.5216/ENG.ebook.978-85-495-0512-5/2022>

Dados Internacionais de Catalogação na Publicação (CIP)
GPT/BC/UFG

E57 Engenharia de software na saúde digital [Ebook] /
organizadores, Fábio Nogueira de Lucena ... [et. al.]. -
Dados eletrônicos (1 arquivo : PDF). - Goiânia : Cegraf
UFG, 2022.

Inclui bibliografia.
ISBN: 978-85-495-0512-5

1. Telemedicina. 2. Gestão de Ciência, Tecnologia e
Inovação em Saúde. 3. Software. I. Lucena, Fábio Nogueira
de II. Programa Educacional em Saúde Digital da
Universidade Federal de Goiás. III. Brasil. Ministério da
Saúde.

CDU: 614:004.41

Bibliotecária responsável: Enderson Medeiros / CRB1: 2276

Engenharia de software na saúde digital

Instituição responsável

Universidade Federal de Goiás (UFG)

Comissão de Governança da Informação em Saúde da UFG (CGIS-UFG)

Centro de Inovação em Gestão da Educação e do Trabalho em Saúde (CIGETS)

Laboratório de Pesquisa em Empreendedorismo e Inovação da Universidade Federal de Goiás (LAPEI-UFG)

Instituição financiadora

Ministério da Saúde (MS)

Secretaria Executiva (SE)

Departamento de Informática do Sistema Único de Saúde (DATASUS)

Secretaria de Gestão do Trabalho e da Educação na Saúde (SGTES)

Apoio

Ministério da Saúde (MS):

Secretaria de Atenção Primária à Saúde (SAPS)

Demais parceiros

Laboratório de Inovação e Estratégia em Governo (LineGov)



Abreviaturas e Siglas

BPMN	<i>Business Process Model and Notation</i>
CGIS	Comissão de Governança da Informação em Saúde
CIGETS	Centro de Inovação em Gestão da Educação e do Trabalho em Saúde
INCA	Instituto Nacional do Câncer
IEC	<i>the International Electrotechnical Commission</i>
ISO	<i>the International Electrotechnical Commission</i>
LAPEI	Laboratório de Pesquisa em Empreendedorismo e Inovação
LineGov	Laboratório de Inovação e Estratégia em Governo
MAPA	Ministério da Agricultura, Pecuária e Abastecimento
RNDS	Rede Nacional de Dados em Saúde
SAPS	Secretaria de Atenção Primária à Saúde
SGTES	Secretaria de Gestão do Trabalho e da Educação na Saúde
SIS	Sistema de Informação em Saúde
SISBOV	Sistema Brasileiro de Identificação e Certificação de Bovinos e Bubalinos
SUS	Sistema Único de Saúde
TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicação
UFG	Universidade Federal de Goiás
UNA-SUS	Universidade Aberta do Sistema Único de Saúde
UnB	Universidade de Brasília



Lista de Figuras e Videoaulas

Figura 1 - Subdomínios do Sistema Único de Saúde (SUS)	12
Figura 2 - Foco na regulação ambulatorial	13
Figura 3 - Perguntas relacionadas à palavra “cigarro”	15
Figura 4 - Resposta à pergunta “Qual o mal que o cigarro faz?”	15
Figura 5 - Principais linguagens de programação, de acordo com o índice <i>TIOBE Programming Community</i>	23
Figura 6 - Mapa mental que reúne alguns modelos de ciclo de vida do desenvolvimento de <i>software</i>	25
Figura 7 - Ciclos de vida de um projeto de <i>software</i>	26
Figura 8 - A casa de madeira mais alta da Rússia	28
Figura 9 - Processo para a elaboração de uma proposta de <i>software</i> integrado à Rede Nacional de Dados em Saúde (RNDS)	40
Videoaula 1 - Como funciona o desenvolvimento de <i>softwares</i> ?	14
Videoaula 2 - Gerência de projeto e modelos de ciclo de vida de <i>software</i>	26
Videoaula 3 - Documentação ágil de requisitos de <i>software</i>	36



Sumário

Apresentação	10
Unidade 1: Contexto de Software	11
1.1 Contexto	12
1.1.1 Sistema de Informação	14
1.2 O que Preciso Saber sobre Desenvolvimento de Software?	14
1.2.1 São Fabricados por Seres Humanos	15
1.2.2 Área de Conhecimento	18
1.2.3 Software Movimenta um Rico Mercado	18
1.2.4 Software é Fundamental para o Mundo Moderno	18
1.2.5 Profissionais de Saúde Devem Participar	18
Unidade 2: Processos de Software	21
2.1 Visão Geral de Processos de Software	22
2.2 Ciclo de Vida de Projeto	24
2.3 Construção Civil Usando Abordagem Incremental	27
2.4 Questões Pertinentes	29
2.4.1 Como Expandir a Capacidade de um Sistema?	29
2.4.2 Qual a Qualidade do Sistema?	30
Unidade 3: Elaboração de Proposta de Software Integrado à Rede Nacional de Dados em Saúde	32
3.1 Roteiro Minimalista	33
3.1.1 Sua Ideia e Riscos	34
3.1.2 Qual a Melhoria Introduzida pela Sua Ideia?	34
3.1.3 Processos	35
3.1.4 Requisitos	35
3.2 Considerações sobre Sua Proposta	37
Unidade 4: Análise, Revisão e Apresentação das Propostas	39
4.1 Registro da Proposta	41
4.2 Análise pelos Pares	41
4.3 Revisão	42
4.4 Eleição	42
4.5 Divulgação do resultado	42
Unidade 5: Encerramento do Microcurso	43
Referências	45



Apresentação

Prezado(a) Participante,

Seja bem-vindo(a) ao Microcurso **Engenharia de Software na Saúde Digital!**

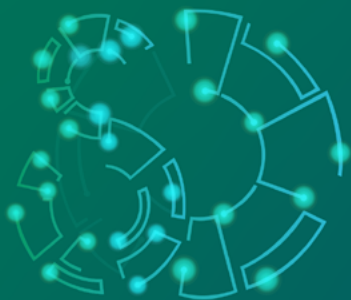
A Saúde Digital depende, em boa parte, do emprego de sistemas de *software* ou, como também são chamados, de programas de computador. O desenvolvimento desses programas é fruto do emprego da Engenharia de *Software*. Naturalmente, neste Microcurso, não há espaço para cobrir com certa amplitude a Engenharia de *Software*, nem tampouco a intenção é formar profissionais para desenvolvimento de *software*.

Essa área do conhecimento é abordada, neste Microcurso, da perspectiva de profissionais da área da saúde, ressaltando aspectos relevantes para que possam contribuir com iniciativas que envolvam a produção de *software* no contexto da saúde. Um resultado natural é o “desconforto” causado pelo uso de termos de um vocabulário desconhecido pelos profissionais da área da saúde. Mais difícil ainda é quando termos são empregados com conotações diferentes, como é o caso de “protocolo”, que possui um uso na saúde distinto daquele da Engenharia de *Software*.

Enfim, a intenção é que profissionais e gestores da saúde sintam-se confortáveis com alguns tópicos úteis à transformação digital da saúde, em particular, no contexto da Estratégia de Saúde Digital do Brasil 2020-2028.

Bom estudo!!!





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Engenharia de software na saúde digital

Unidade 1 **Contexto de Software**

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior



Unidade 1: Contexto de Software

1.1 Contexto

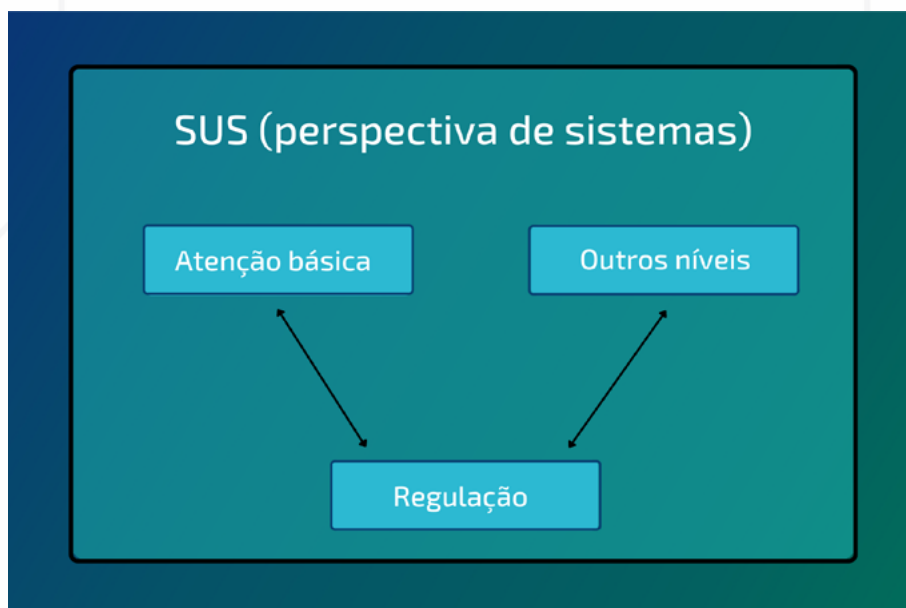
Na área da tecnologia da informação e comunicação (TIC) o termo *software* faz parte da essência da área e é utilizado corriqueiramente. Embora empregado com uma certa constância, o que é? Como é concebido? Qual a sua finalidade?

Software é produzido ou atualizado para atender demandas de um dado domínio. Mas o que é domínio?

Domínio¹ é o nome que se dá à área de atuação do *software*. Vejamos o exemplo a seguir. A saúde é um domínio. Nesse extenso domínio, podemos nos restringir ao que habitualmente denominamos de Sistema Único de Saúde (SUS), o que é um domínio mais restrito. O domínio SUS, por sua vez, pode ser subdividido de várias formas, em domínios ainda mais específicos. Uma delas é ilustrada na Figura 1, formada por três subdomínios:

- Atenção básica;
- Outros níveis; e
- Regulação.

Figura 1 - Subdomínios do Sistema Único de Saúde (SUS)



Fonte: autoria própria.

Não se assuste! A Figura 1 não representa uma proposta de mudança na organização do SUS. Apenas ilustra uma perspectiva de interesse, nesse caso, sistemas de informação em saúde que atendem a “Atenção básica”, “Outros níveis” e a “Regulação”. Ou seja, mergulhamos na saúde, um escopo amplo, e identificamos um domínio mais restrito, o SUS, e não todo o SUS, mas, apenas os sistemas de informação que apoiam o SUS. Sistemas de informação são definidos na seção seguinte.



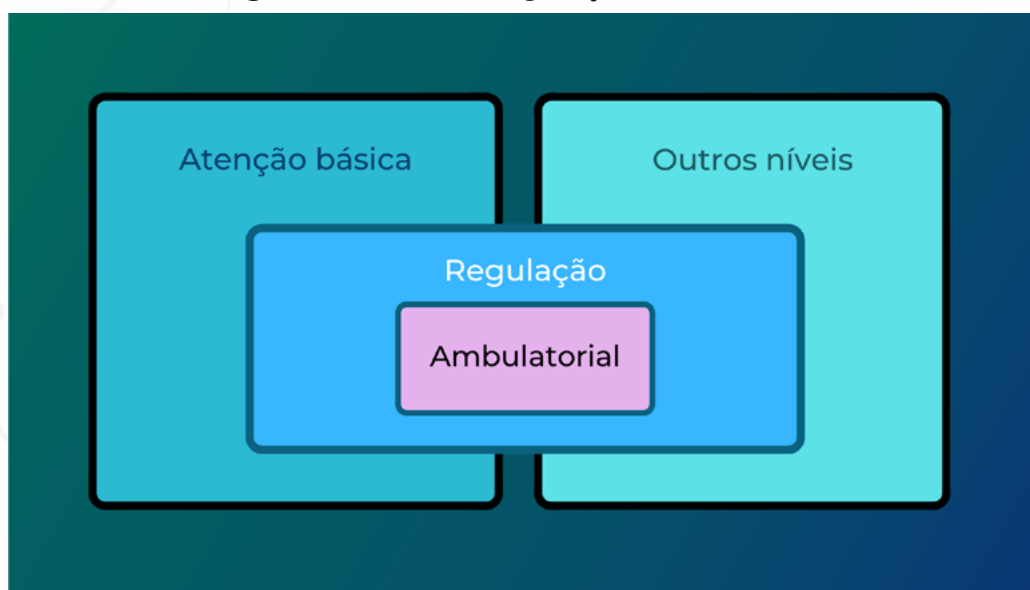
Abaixo (Figura 2), temos um refinamento da Figura 1, na qual fica claro que nem todo o domínio do complexo regulador é de interesse, mas, uma parte desse, o domínio da Regulação Ambulatorial. E por que fazer esses refinamentos sucessivos? No contexto desta Unidade de ensino, o objetivo é identificar, conforme já mencionado, a área específica de atuação de um *software*. Se estamos interessados em propor um *software* que vise a atender a Regulação Ambulatorial, então, esse domínio é uma forma de delimitar a parte de interesse da saúde.

O complexo regulador está organizado em:

- a. Central de regulação de consultas e exames;
- b. Central de regulação de urgências; e
- c. Central de regulação de internações hospitalares.

Consulte a Política Nacional de Regulação do SUS para mais detalhes.

Figura 2 - Foco na regulação ambulatorial



Fonte: autoria própria.

Antes que o *software* seja construído para atender as demandas de um domínio, é comum buscar uma compreensão de como esse domínio “funciona”, quais os processos pertinentes, quais os objetivos, o que é relevante, quais as pessoas envolvidas, quais os papéis, quais as instituições, quais as informações que fluem, quais os fluxos dessas informações, quem financia, quem opera e assim por diante.

Em geral, uma percepção clara de um domínio de interesse facilita a identificação de possíveis melhorias e evoluções que podem ser introduzidas nesse domínio, por exemplo, por meio da introdução de sistemas de informação, ou da modificação de sistemas existentes. Quando um sistema é introduzido ou modificado, convém ressaltar, atribuições e tarefas migram, em geral, de pessoas para sistemas de informação, o que altera os processos. Conforme Mani e Ebert (2022),² *software* é constituído de pessoas e tecnologia.

É no contexto de um domínio que um *software* é pensado, criado ou modificado. E a introdução de um *software* ou a modificação de um existente visa a contribuir com os objetivos e resultados desse domínio. Por fim, para alterar resultados será necessário, quase que invariavelmente, modificar os processos empregados.



A introdução de interoperabilidade entre o sistema empregado por um laboratório e o sistema empregado por um outro estabelecimento de saúde, por exemplo, pode eliminar a necessidade de transporte físico de laudos de exames entre eles. Para isso, os laudos precisam ser assinados digitalmente, o que, seguramente, irá interferir nos processos tanto do laboratório quanto do estabelecimento de saúde em questão. Essa introdução é um exemplo de transformação digital, visando agilizar o acesso à informação relevante para o cuidado do paciente assistido e reduzir custos, além de simplificar processos de negócio.

1.1.1 Sistema de Informação

Conforme artigo da Wikipédia,³ sistema de informação é um sistema projetado para coletar, armazenar e distribuir informações, composto por tarefas, pessoas, papéis e tecnologia. Um Sistema de Informação em Saúde (SIS) é um sistema de informação no domínio da saúde.

A Engenharia de *Software* é a área do conhecimento que reúne o conhecimento dedicado à produção de *software* (tecnologia). Em particular, não estamos interessados em todo tipo de *software*, mas *software* empregado no escopo da saúde, ou seja, *software* que é parte de um SIS.

1.2 O que Preciso Saber sobre Desenvolvimento de *Software*?

Não é intenção deste Microcurso formar engenheiros de *software*, mas, capacitar profissionais para compreender algumas questões pertinentes dessa área e, em consequência, habilitá-los para contribuir com esforços envolvendo *software* em seus domínios de atuação. Conforme definição anterior, o mais correto seria dizer esforços de definição e desenvolvimento de sistemas de informação em saúde, o que inclui *software* sim, mas também tarefas, pessoas e papéis.

Assista, a seguir, a Videoaula 1 que aborda uma visão geral sobre as atividades necessárias para o desenvolvimento de *software*, ministrada pelo Prof. Dr. Fábio Nogueira de Lucena.

Videoaula 1 - Como funciona o desenvolvimento de *softwares*?



Fonte: autoria própria.



LINK

1.2.1 São Fabricados por Seres Humanos

Parece óbvio, todo mundo sabe. Talvez menos perceptível sejam as consequências do óbvio. A seguir, um cenário ilustra o impacto do ser humano em resultados que, em muitos casos, são atribuídos a um *software*, a um sistema, como se fossem entidades autônomas. Quando alguém faz uma busca por determinada informação na *Internet*, digamos, usando o serviço de busca da Google®, muitas vezes as respostas são numerosas, centenas, milhares e até milhões de resultados. Usando todo o preconceito possível, esteja avisado, a ordem dos resultados não será “aleatória”. É bem provável que o que é “mais pertinente” à consulta seja apresentado primeiro, sim, é possível que isso seja feito. Em geral, contudo, ainda teremos muitos resultados pertinentes, todos “igualmente válidos”. Nesse caso, como ordená-los tendo em vista que os usuários tendem a consultar “apenas os primeiros da lista de resultados”?

Em uma consulta com uma única palavra “cigarro”, a página de resposta diz que há mais de 39 milhões de respostas. Também lista perguntas atribuídas a pessoas, conforme ilustrado a seguir. Nesse excerto, foram selecionadas apenas as quatro primeiras perguntas que aqui são tratadas como títulos “alternativos” para os reais títulos das páginas subjacentes.¹

Figura 3 - Perguntas relacionadas à palavra “cigarro”

As pessoas também perguntam

Qual o mal que o cigarro faz?



Quais são os benefícios do cigarro?



Por que o cigarro vicia?



Quais são as características do cigarro?



[Feedback](#)

Fonte: Google®.

Se formos classificar as perguntas acima em “contrárias”, “favoráveis” e “neutras” ao consumo do cigarro, provavelmente teremos respostas variadas. Agora vamos aos detalhes. Quando consultamos a primeira pergunta, retorna o resultado abaixo.

Figura 4 - Resposta à pergunta “Qual o mal que o cigarro faz?”

Qual o mal que o cigarro faz?



O tabagismo é uma doença (dependência de nicotina) que tem relação com aproximadamente 50 enfermidades, dentre elas vários tipos de câncer (pulmão, laringe, faringe, esôfago, estômago, pâncreas, fígado, rim, bexiga, colo de útero, leucemia), doenças do aparelho respiratório (enfisema pulmonar, bronquite crônica, asma, ...

<https://www.inca.gov.br> > perguntas-frequentes > quais-... ▼

Quais são as doenças causadas pelo uso do cigarro e outros ...

Fonte: Google®.

Sem receio de sermos preconceituosos, conforme alertado acima, ficamos com a impressão de que o Instituto Nacional do Câncer (INCA), no título original do artigo em questão, “Quais são as doenças causadas pelo uso do cigarro e outros...” não deixa dúvidas de que é “contrário” ao consumo de cigarro. O termo “contrário” foi introduzido na classificação sugerida acima. Por outro lado, a pergunta apresentada pelo Google® “Qual o mal que o cigarro faz?” é um eufemismo se comparado com o título do artigo do INCA. Novamente, já antecipamos o nosso preconceito, mas você pode esquecer dele e fazer o seu próprio julgamento dos títulos e identificar se são títulos equivalentes ou se caíam em categorias distintas conforme a classificação fornecida acima.

Se fizermos uma análise das questões apresentadas pelo Google®, por exemplo, a segunda delas, “Quais são os benefícios do cigarro?”, conforme a lista acima (Figura 3), dificilmente pode-se afirmar que esta pergunta é “neutra”. É mais provável que seja considerada “favorável” ao consumo de cigarro.

Por fim, observe a última pergunta “alternativa” da lista: “Quem tem crise de ansiedade pode fumar?”. Segundo nosso preconceito, a pergunta sugere uma possibilidade, uma dúvida. O título original, contudo, é “Parar de fumar ajuda a controlar a ansiedade”. Novamente, nosso preconceito nos diz que o título original deixa claro o benefício de se parar de fumar, enquanto o título “alternativo”, produzido pela resposta, sugere uma dúvida.

A intenção aqui não é defender ou condenar o uso do cigarro, nem tampouco uma ou outra empresa que ofereça serviços de busca. Inclusive alertamos que seríamos preconceituosos. A intenção é deixar claro que *software* não é necessariamente “neutro”, ou está do “lado bom” de uma causa simplesmente por ser *software*. Enquanto ferramenta, *software* pode ser utilizado para boas finalidades e outras nem tanto. *Software* não tem preconceito, contudo, os humanos que o produzem têm.

Seguem algumas questões para a sua reflexão.

Você acredita que um
“título alternativo” pode
sugerir algo diferente do
título real de um
documento?

Você acredita que um
software dedicado a gerar
títulos alternativos pode ser
orientado a selecionar títulos
mais “favoráveis” ou mais
“contrários” a uma dada
causa?



Você acredita que é possível ser “neutro” ao listar um resultado de uma consulta na qual estão incluídas centenas de respostas, produzidas pela academia, empresas privadas, grupos políticos, religiosos e outros?

Você acredita que a lista das respostas, as simples respostas, mesmo sem qualquer “intervenção” como títulos “alternativos” possa ser “neutra”?

Independentemente da neutralidade ou não, você acredita que a lista de respostas pode ser montada para aderir a uma determinada linha de pensamento?



1.2.2 Área de Conhecimento

O conhecimento pertinente ao desenvolvimento e manutenção de *software* é reunido na área denominada Engenharia de *Software*. Essa área compila propostas de soluções para as demandas crescentes do emprego de *software*. Não há um portal definitivo nem completo sobre Engenharia de *Software*, mas você poderá “saborear” o que de mais relevante ocorre nesse domínio em portais como InfoQ⁴ e *Software Engineering Radio*.⁵ Por fim, o corpo de conhecimento dessa área está bem identificado em *Software Engineering Body of Knowledge*, disponível gratuitamente.⁶

1.2.3 Software Movimenta um Rico Mercado

As demandas por *software* têm movimentado um rico mercado. Observe que as cinco pessoas mais ricas do planeta são, nesta ordem²: Elon Musk (Tesla®), Jeff Bezos (Amazon®), Bernard Arnault (Christian Dior®), Bill Gates (Microsoft®) e Larry Page (Google®).⁷ Pode-se afirmar que todos dependem de *software* e, em particular, quatro desses têm seus negócios fortemente baseados em *software*. Isso fica ainda mais evidente se considerarmos as empresas de maior valor de mercado, as cinco maiores, em ordem decrescente: Apple®, Microsoft®, Amazon®, Alphabet® e Meta®. Talvez deva ser mencionado que Alphabet® é conhecida por Google® e Meta reúne Facebook®, Instagram® e WhatsApp®.⁸

1.2.4 Software é Fundamental para o Mundo Moderno

O mundo como o conhecemos hoje depende fortemente de *software*. Se removermos a dependência, teremos um retrocesso em termos de eficiência e eficácia em muitas atividades e, talvez, até a impossibilidade de realizar algumas delas.

Na saúde não é diferente. O emprego de *software* é imprescindível para substituir o uso de papel e tal substituição está associada a vários benefícios. Muitas vezes, nos dedicamos a discutir registro eletrônico e prontuário eletrônico, por exemplo, como se isso fosse um fim em si. Não é. É parte de um esforço no qual o *software* assume um papel importante em processos da saúde. A Saúde Digital é, em parte, a assistência à saúde revisitada e apoiada pela introdução de SIS.

1.2.5 Profissionais de Saúde Devem Participar

Se você pretende contribuir diretamente com o desenvolvimento de SIS, então será preciso esforço considerável para adquirir conhecimentos e habilidades em Engenharia de *Software*. Naturalmente, isso está além do escopo do presente Microcurso. O restante do documento usa uma perspectiva diferente, a perspectiva de profissionais que estarão envolvidos, mas que não desempenham atividades típicas de engenheiros de *software*.

Tomemos como referência os profissionais da Enfermagem. Eles são (ou serão) usuários de SIS. Quando um sistema é introduzido em um domínio no qual fazem parte, esse domínio é “alterado”, o que também altera os papéis a serem realizados por eles. Afinal, já sabemos que um sistema de informação inclui tarefas, pessoas e papéis, além de tecnologia. Nessa introdução é natural que processos sejam modificados. Por exemplo, em um instante do passado era necessário **imprimir, reunir e despachar impressos** para um laboratório ou, no sentido inverso, **receber laudos impressos e distribuí-los em pastas suspensas**, o que é modificado com a adoção de um sistema de informação, que permite clicar botões para **enviar requisição de exame** e **confirmar recebimento de laudo**, por exemplo. Novamente, mudanças como essas reordenam como as “coisas” são feitas em um estabelecimento de saúde, e isso repercute nos profissionais envolvidos.

Tendo em vista o parágrafo anterior, quem melhor poderia contribuir com a definição dos novos processos de um estabelecimento de saúde a serem apoiados por um sistema de informação? Se os profissionais da Enfermagem são atores dos processos correntes e serão daqueles futuros, então, eles naturalmente devem ser envolvidos durante a concepção dos novos processos, assim como outros profissionais. Esses atores também precisam contribuir com a definição do papel a ser atribuído ao SIS que será introduzido ou modificado. Esse papel é conhecido, normalmente, por funções ou requisitos do sistema de informação.⁹

Além da definição do novo *software* ou da adaptação de um existente, é bom que tais profissionais tenham uma noção de como isso é realizado por engenheiros de *software* (processos de desenvolvimento de *software*), até porque, apenas no mundo ideal, os interesses dos usuários podem ser atendidos em sua totalidade. No mundo real, existem restrições como custo e tempo, por exemplo. Existem muitas outras restrições, inclusive introduzidas pela tecnologia, inclusive pela própria Engenharia de *Software*.

Se vamos participar do processo de construção (desenvolvimento) de um *software*, se estamos envolvidos na transformação digital da saúde, então, não é exigido que sejamos engenheiros, mas teremos que conhecer um pouco sobre a Engenharia de *Software*, assim como aprenderemos um pouco da Engenharia Civil se participarmos da construção de um prédio.

Doravante, o conteúdo apresentado visa colaborar com o conhecimento dos profissionais da área da saúde a respeito de informações pertinentes a tecnologias empregadas no desenvolvimento de *software*.

1.3 Quiz

Para testar os conhecimentos adquiridos até aqui, responda ao *quiz* no Ambiente Virtual.

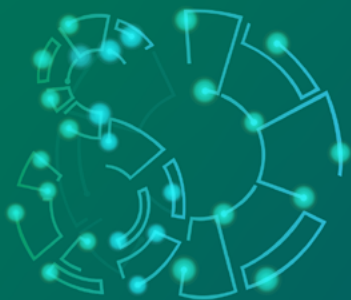




Para lembrar...

- Sistema é um conjunto de elementos que interagem entre si, estão inter-relacionados e, dessa forma, definem um todo. Consulte a [Wikipedia](#) para detalhes.¹⁰
- Um grande sistema é o Sistema Único de Saúde (SUS). Também há o sistema nervoso, o sistema respiratório e outros conhecidos na saúde. Também pode-se citar o sistema energético brasileiro, o sistema elétrico de um automóvel, o sistema hidráulico de um clube de recreação, um sistema de irrigação, um sistema de manejo de gado. Também temos sistemas de informação.
- Um software que faz parte de um sistema de informação também é considerado, por si só, um sistema de software, tendo em vista a complexidade típica desse tipo de “objeto”. Enfim, sistemas existem em todos os lugares, sejam criados pelo ser humano ou não, como o nosso sistema solar.





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Engenharia de software na saúde digital

Unidade 2 **Processos de Software**

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior
Renata Dutra Braga
Taciana Novo Kudo



Unidade 2: Processos de Software

2.1 Visão Geral de Processos de Software

Processo de *software* é a denominação do que é feito para produzir *software*. Produzir, nesse contexto, inclui adaptações, correções e a construção propriamente dita. Seria bom se existisse um processo infalível para produzir *software*. Bastaria segui-lo para obter o resultado esperado. O fato é que a noção de processo ideal é utópica no presente.

Existe um conjunto significativo de variáveis (fatores) que interferem na definição de um processo de *software*, tanto que o comum é abordar não um processo de *software* específico, mas, abstrações ou modelos de processos de *software*, o que é feito nesta Unidade. Quando se tem uma necessidade a ser atendida por um dado *software*, deve ocorrer a seleção e adaptação de um ou mais modelos que devem ser combinados, à luz do *software* e das restrições do projeto em questão. O restante do texto fornece elementos que ajudam a elucidar esse parágrafo.

Conforme Pressman e Maxim (2020),¹¹ subjacente a todo processo de *software* há um conjunto de atividades aplicáveis a todos os cenários. Ou seja, essas atividades são realizadas de uma forma ou de outra, após as adaptações, mas sobrevivem a elas.

Tendo em vista nosso escopo aqui neste Microcurso, vamos nos concentrar nessas atividades mais abstratas, genéricas, em vez de tarefas específicas e pontuais, que são bem mais técnicas e suscetíveis ao contexto, além de ser de interesse de profissionais envolvidos tecnicamente com a produção de *software*.

As atividades definidas são: comunicação, planejamento, modelagem, construção e implantação.¹¹

Comunicação. Em geral, um projeto de *software* envolve várias pessoas e instituições, também chamados de interessados (*stakeholders*). No contexto de um *software* usado por um determinado município, apesar de inúmeras variações, são vários os interessados: os profissionais que vão fazer uso do *software* nos estabelecimentos de saúde; os indivíduos assistidos (se recebem alguma notificação, por exemplo, aviso de “consulta confirmada” ou outra); o prefeito que terá que “abraçar o projeto” e selecioná-lo dentre as demandas existentes. Há, ainda, questões de infraestrutura (rede, computadores, *Internet*); há questões de interoperabilidade (integração com a Rede Nacional de Dados em Saúde [RNDS]); há os profissionais que irão produzir o *software*; há aqueles que irão mantê-lo e outros. Esse conjunto de pessoas e de grupos interessados só irá se entender se houver esforço significativo e intencional, visando a uma “boa comunicação”.

Planejamento. Em 1950, o general Dwight Eisenhower escreve uma carta na qual se lê: “Planos produzidos em tempo de paz são inúteis, mas o planejamento em tempo de paz é indispensável”.¹² Isso se popularizou de várias formas, talvez a mais comum seja: “Planos são inúteis, mas planejamento é tudo”. Além da curiosidade histórica, serve para esclarecer que planos são esforços para caracterizar e interferir no futuro. Mesmo assim, podemos nos preparar ou atuar no presente, visando a uma situação desejada. E dado que o contexto muda com frequência, é preciso planejamento constante. Conforme Pressman e Maxim (2020),¹¹ esse plano é um mapa que guia a equipe, todos os envolvidos, em uma jornada complicada. Esse mapa é registrado em um Plano de Projeto de *Software*.



Modelagem. Profissionais de saúde e gestores criam modelos de funcionamento de centrais de regulação, quando definem processos de negócio a serem observados por reguladores e outros. Profissionais de saúde também criam modelos de assistência, quando desenham fluxogramas ou diagramas em notação *Business Process Model and Notation* (BPMN)¹³ para orientar a forma de recepcionar (triar) um indivíduo com uma dada queixa. Na construção de *software*, modelos permitem: analisar requisitos; registrar a arquitetura do *software*; o projeto detalhado do *software*; a interface com o usuário; os detalhes de implementação do *software*; a operação do *software*; a infraestrutura empregada e outros. Talvez não exista outra área que faça tanto uso de modelos quanto a Engenharia de *Software*, e há uma hipótese razoável para essa afirmação: *software* é uma entidade “abstrata”, nunca “vista” por ninguém, mas apenas “percebida” pela manifestação da sua execução e, em consequência, a produção de *software* é essencialmente produzir modelos.

Construção. A construção pode ser eleita como a atividade mais concreta associada ao abstrato *software*. Construção é a atividade que cria código segundo uma linguagem de programação, testes e outros artefatos necessários para a execução em um *hardware*. Você já deve ter ouvido falar em linguagens de programação como C, Java, C#, JavaScript e outras, e até HTML, CSS e assim por diante. Pois bem, são algumas das linguagens empregadas na construção de *software*. Java, por exemplo, é apenas uma dentre centenas de linguagens de programação.¹⁴ A escolha da linguagem de programação depende do projeto, depende do *software* a ser desenvolvido e de vários outros fatores. Contudo, historicamente, há uma prevalência de algumas linguagens (Figura 5).¹⁵

Figura 5 - Principais linguagens de programação, de acordo com o índice *TIOBE Programming Community*



Fonte: Adaptado de TIOBE.¹⁵



Implantação. Essa atividade coloca o *software* à disposição de usuários finais, aqueles com os quais o *software* irá interagir. A implantação também envolve, em geral, treinamento de usuários e aquisição de *hardware*, por exemplo. Convém destacar, apenas por uma questão de rigor, que nem todo *software* oferece interface para usuários humanos. Há *software* que interage apenas com *hardware* e/ou outro *software*.

Essas atividades são realizadas de forma específica para cada cenário. Conforme mencionado anteriormente, os detalhes vão variar conforme o contexto. Por exemplo, a comunicação pode se beneficiar de aplicativos como Whatsapp, ou ser realizada formalmente por meio de documentos registrados em cartório. A decisão, nesse caso, pode considerar as preferências dos interessados, leis a serem observadas e outros.

O planejamento de um sistema crítico (executado em aviões, dispositivos da área da saúde e outros) é diferente de um sistema cujos “danos são limitados” mesmo na presença de mau funcionamento. Um sistema crítico exige o emprego de modelos ditos “formais”, que são modelos cuja interpretação semântica está baseada no rigor matemático. Por exemplo, Petri Net¹⁶ é uma notação formal e empregada para registrar atividades concorrentes, o que também pode ser feito usando a BPMN.¹³

A linguagem de programação, que é a linguagem usada por desenvolvedores para produzir código que computadores executam, recebe forte influência de fatores relevantes para o produto e, talvez, até para o projeto. Por exemplo, caso o desempenho (rapidez na execução do *software*) seja um fator primordial para o sucesso, é possível que a linguagem de programação C seja adotada. Em consequência, a equipe terá uma produtividade inferior àquela de uma equipe que adote Java ou C#. Se o projeto em questão, não o produto (*software*), não admite essa produtividade inferior, então, é possível que a escolha da linguagem seja influenciada não pelo produto, mas pelo projeto.

Os itens ilustrados acima são apenas alguns dentre vários que impactam o produto e o projeto em questão. Em tempo, tais itens estão além do escopo deste Microcurso. São detalhes relevantes para profissionais de *software*. Tendo em vista o nosso escopo, perceba que projetos de *software* exigem um conjunto considerável de decisões, algumas pertinentes ao projeto e outras ao produto, onde é natural que uma delas esteja mais alinhada ao contexto em questão do que outra.

2.2 Ciclo de Vida de Projeto

As atividades pertinentes a processos de *software* são executadas, em geral, ao longo de um projeto. Por exemplo, o Projeto X no qual é desenvolvido o *software* para atender determinada demanda de um estabelecimento de saúde.

Um projeto é um esforço com início e fim bem definidos voltado para a produção de um produto ou serviço (único). Convém ressaltar que podem ser necessários vários projetos para produzir um *software* ou um projeto pode incluir todo o desenvolvimento de um *software*.

O que é necessário para produzir um *software* foi apresentado na seção anterior. O que nos interessa nesta seção é o que ocorre ao longo de um projeto de *software*. Em um projeto que visa ao desenvolvimento de *software*, as atividades da seção anterior são executadas, mas além

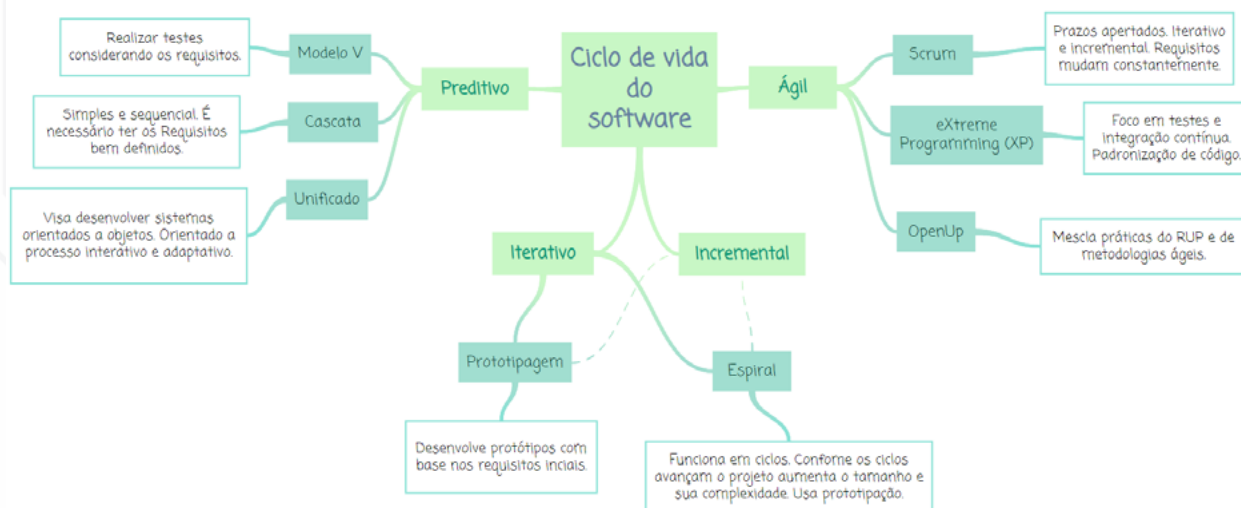


delas, e considerando o projeto, e não apenas o *software* por ele produzido, há outras atividades, as atividades pertinentes ao gerenciamento do projeto. Essas atividades estão organizadas em fases que compõem o ciclo de vida de um projeto.

São identificados quatro tipos de ciclo de vida (Figura 6):

- a. preditivo;
- b. iterativo;
- c. incremental e
- d. ágil.¹⁷

Figura 6 - Mapa mental que reúne alguns modelos de ciclo de vida do desenvolvimento de *software*



Fonte: autoria própria.

O uso de um tipo de ciclo de vida ou outro estilo ou a combinação deles é definida por projeto, visando o sucesso do projeto. Novamente, a aderência ao contexto tem repercussões. O processo de escolha provavelmente é similar à prescrição de um medicamento, na qual o profissional de saúde avalia o cenário e identifica o que é mais adequado.

- **Preditivo.** Uma abordagem mais tradicional, também referenciada como “pesada” (*heavyweight*), burocrática. Nesse tipo, a maioria do planejamento ocorre logo no início do projeto, que é executado como uma sequência de fases, em um único ciclo.
- **Iterativo.** Nesse tipo, prioriza-se a possibilidade de avaliar o trabalho incompleto visando a identificação de melhorias e possíveis mudanças. O esforço é realizado em vários ciclos.
- **Incremental.** O *software* é desenvolvido em ciclos e cada um deles oferece uma versão pronta para uso.
- **Ágil.** Combina iterativo e incremental, visando a entregas frequentes.

As características desses tipos estão ilustradas na Figura 7.



Figura 7 - Ciclos de vida de um projeto de *software*

CARACTERÍSTICAS				
ABORDAGEM	REQUISITOS	ATIVIDADES	ENTREGA	OBJETIVO
Preditiva	Fixos	Realizadas uma vez por projeto	Única	Gerenciar custos
Iterativa	Dinâmicos	Repetidas até estar correto	Única	Correção da solução
Incremental	Dinâmicos	Realizadas uma vez por incremento	Frequentes	Velocidade
Ágil	Dinâmicos	Repetidas até estar correto	Frequentes	Gerar valor

Fonte: Adaptada de PMI (2017).¹⁷

Assista, a seguir, a Videoaula 2 que trata sobre gerência e modelos de ciclos de vida para desenvolvimento de um projeto de *software*, ministrada pela Prof^a. Dra. Taciana Novo Kudo.

Videoaula 2 - Gerência de projeto e modelos de ciclo de vida de *software*



Fonte: autoria própria.



LINK

Convém fornecer alguns esclarecimentos sobre o conteúdo da Figura 7 e, na sequência, comentar um pouco mais sobre a abordagem incremental, aquela cujo objetivo é a velocidade. O que todos nós desejamos, certo? Sim, mas, também desejamos uma solução correta, também desejamos entregar valor e, por fim, quem não se interessa pela gestão dos custos?



A abordagem preditiva é adequada para cenários onde os requisitos são conhecidos ou estão predefinidos. O cliente do *software* sabe o que o *software* precisa fazer. Isso significa que o conjunto de funções atribuídas ao *software* é estável e bem conhecido. Outros cenários, por outro lado, possuem incertezas, dúvidas que nem os interessados são capazes de esclarecer sem um significativo esforço. Nesses casos, não há como adotar a abordagem preditiva. Isso significa que a gestão de custos torna-se um fator de cuidado adicional, afinal, o escopo do *software* não está caracterizado em detalhes.

A abordagem iterativa premia a qualidade. Nela, vamos iterar até que o *software* esteja bom. Quando isso ocorrer, o *software* é entregue ao cliente. Observe que, excetuando-se a abordagem preditiva, as demais lidam com incertezas. Essas incertezas podem estar nos requisitos (escopo), que só serão conhecidos ao longo do tempo, com o andamento do projeto, ou na tecnologia utilizada. O *software* pode não atender adequadamente flutuações de demanda. Pode estar limitado a uma certa capacidade de produção. Enfim, *software*, assim como qualquer outro artefato produzido pelo ser humano, tem limitações, e a abordagem iterativa é aquela que visa a reduzi-las a cada iteração.

Conforme mencionado, exceto a abordagem preditiva, as demais são executadas em vários ciclos. Ou seja, teremos que nos comunicar, planejar, modelar, construir e inclusive implantar “partes” do *software* a cada ciclo. E repetir tudo ao executar o próximo ciclo. Nessa abordagem, portanto, teremos vários momentos de planejamento, por exemplo, em vez de um único momento onde o planejamento é realizado (caso da abordagem preditiva).

A abordagem incremental é apresentada no contexto de um exemplo (analogia) com a construção civil na seção seguinte. Não interprete como a “principal” ou “melhor”, pois, nesta altura, já deve estar claro que tal conceito é “estranho” porque ignora o contexto. E a última é uma combinação das abordagens iterativa e incremental^{18,19}.

2.3 Construção Civil Usando Abordagem Incremental

Abordagem incremental é algo do contexto de projeto de *software* e não do contexto de projetos da construção civil. Contudo, feito o aviso, ninguém irá nos impedir de fazer o uso em outro escopo que não o original.

Na abordagem incremental o resultado produzido em cada ciclo é acrescido ao que existia anteriormente e disponibilizado para uso. Nesse caso, os usuários não esperam a conclusão de todo o *software* para que possam fazer uso (preditiva), ou até que a qualidade de tudo esteja aprovada para uso (iterativa). O conjunto de funcionalidades é acrescido de uma nova funcionalidade (incremento) e o resultado dessa soma disponível para uso. Isso é ilustrado na Figura 8 por meio da construção civil, a casa de madeira mais alta da Rússia, construída ao longo de 15 anos, vários ciclos, sem planejamento formal ou permissão legal.²⁰



Figura 8 - A casa de madeira mais alta da Rússia



Fonte: https://en.wikipedia.org/wiki/Sutyagin_House.²⁰

A ausência de um planejamento amplo e antecipado pode resultar em um *software* equivalente à casa ilustrada. Contudo, esse não necessariamente é o motivo do resultado. Incertezas provavelmente existiram e não foram adequadamente tratadas, os riscos correspondentes foram simplesmente aceitos ou sequer identificados.

De fato, existe o conceito de construção incremental:

processo gradual onde componentes são acrescentados à medida que financiamento, tempo ou materiais tornam-se disponíveis. Desta forma, os custos podem ser reduzidos.²¹

Naturalmente, agora, no sentido inverso, tais benefícios e contexto da construção incremental também são esperados em projetos de *software*.



Há muitas questões aqui envolvidas e esses detalhes serão evitados. O leitor interessado pode consultar outras referências.²² A questão principal aqui é perceber a existência de modelos distintos para se construir *software*.

2.4 Questões Pertinentes

Enquanto membro de uma equipe que contribui com o desenvolvimento de *software*, você terá contato com vários “tópicos”. Alguns deles são comentados abaixo. A intenção não é tornar o leitor um especialista, mas ciente do que se trata, até para habilitá-lo a intervir, se for o caso.

2.4.1 Como Expandir a Capacidade de um Sistema?

Um sistema feito para um posto de saúde, por exemplo, no qual trabalham não mais que 10 profissionais para atender uma demanda “reduzida”, não necessariamente está adequado a um posto de saúde no qual há dezenas de profissionais para atender uma demanda de centenas de pessoas.

Alguém pode dizer que basta adquirir um disco com maior capacidade, ou mais memória, ou computador com maior capacidade de processamento. A esse tipo de extensão chamamos de **escalabilidade vertical**³. Seguramente, pode-se obter melhorias quando se escala verticalmente, contudo, há limitações que só podem ser eliminadas com a **escalabilidade horizontal**⁴.

Quando desenvolvido, um *software* pode não estar preparado para escalar horizontalmente. Observe que a escalabilidade vertical diz respeito ao *hardware* (onde o *software* é executado). Por exemplo, em um computador mais veloz, o mesmo *software* tende a ter um desempenho melhor. Escalar horizontalmente, por outro lado, exige compatibilidade do *software*. O *software* pode não ter sido feito para atender esse requisito. E por que alguém não acrescentaria tal requisito em todo e qualquer *software*? Nem todo *software* precisa escalar horizontalmente e, como é de se esperar, há um custo tanto de projeto quanto de implementação para isso acontecer.

Se você está desenvolvendo um *software* que vai ser testado, por exemplo, no município de Jacundá, no Pará, mas a intenção é de que esse *software* seja utilizado em todo o Estado do Pará, talvez até por todos os municípios brasileiros. Então, esse *software* precisa escalar horizontalmente, mesmo que, para atender a demanda de Jacundá, isso não seja necessário. Ou seja, se o *software* não escalar horizontalmente, ele pode até satisfazer completamente as necessidades de um município, mas provavelmente não estará apto para atender a demanda conjunta de vários municípios.

E não é preciso todo um Estado ou país para se deparar com limitações. Ou seja, um *software* pode atender adequadamente à demanda local de um estabelecimento de saúde com um único usuário e, ao mesmo tempo, não oferecer a possibilidade de ser utilizado simultaneamente por dois ou mais usuários.

Por fim, convém identificar se os objetivos de negócio a serem apoiados por um sistema de informação exigem escalabilidade horizontal. Se afirmativo, então é razoável um esforço de *design* correspondente, focado na escalabilidade horizontal.

3 Escalabilidade vertical: habilidade de aumentar a performance (capacidade) do software, melhorando os seus recursos de infraestrutura (substituir um servidor por outro servidor mais potente).

4 Escalabilidade horizontal: se obtém com o acréscimo de computadores, portanto, a capacidade é ampliada com o acréscimo de nós com capacidade de processamento.

2.4.2 Qual a Qualidade do Sistema?

Quando vamos definir um sistema ou participar da definição de um, naturalmente, surgem requisitos funcionais como “o sistema deve permitir que seja obtida uma versão PDF do resultado de exame”, por exemplo. Os requisitos funcionais são “mais simples”, porque se referem ao comportamento do sistema, aquilo que se espera que ele faça para auxiliar um processo de negócio. Por outro lado, os requisitos não funcionais ou requisitos de qualidade, que considero um nome mais apropriado, são “mais sutis”.

Escalabilidade, assunto tratado na seção anterior, é um exemplo de requisito de qualidade.

Usabilidade é outro exemplo de requisito de qualidade e é facilmente detectável pelo usuário. Quando um usuário “reclama” da dificuldade de se realizar determinada operação com um sistema, da dificuldade de se lembrar de como requisitar determinada operação e assim por diante, está fazendo uma avaliação da qualidade do sistema. Observe que não se trata de fornecer ou não determinada funcionalidade, mas de quão “fácil” é o acesso a essa funcionalidade.

O fato é que existem muitos outros aspectos qualitativos que podem ser utilizados para caracterizar um sistema. Conforme ISO/IEC 25010 (2011),²³ existem dezenas deles. Novamente, detalhes ficam para os técnicos em *software*, alguns deles, contudo, convém esclarecer.

Primeiro, a noção de qualidade. **Qualidade** é amplamente definida como “conformidade com requisitos”. Isso significa que não há como medir a qualidade de um sistema de informação na ausência dos requisitos que foram empregados na definição dele, seja de forma explícita ou não. Isso significa, leitor(a), que você deve estar atento(a) à caracterização de um produto de *software* e não apenas em eventual avaliação desse *software* na perspectiva de qualidade. De fato, conforme essa definição, não há como avaliar a qualidade sem a definição dos requisitos.

Os outros aspectos de qualidade aqui mencionados incluem segurança e disponibilidade. Essa última quase sempre associada com confiabilidade.

Segurança inclui confidencialidade e integridade, o que são aspectos necessários em sistemas que gerenciam informações sensíveis, como as informações em saúde. Naturalmente, há outras questões pertinentes à segurança, por exemplo, não repúdio e autenticação, geralmente implementadas por meio de uso de certificados digitais.

Disponibilidade é medida como o percentual do tempo em que o sistema encontra-se operacional, pronto para uso.

A **confiabilidade**, outro requisito de qualidade, é medida de forma diferente, em geral, por meio do tempo médio entre falhas e da taxa de falhas.

Existem outros tópicos? Conforme já mencionado, há muitos outros.²³ A segurança foi destacada pelo contexto da saúde, sobre o qual há uma atenção especial ao acesso de informações pertinentes aos indivíduos assistidos. A disponibilidade e a confiabilidade são decorrentes da dependência que existe, e está se ampliando, em relação aos SIS. Afinal, tais sistemas não são perfeitos, nem tampouco são executados em um *hardware* perfeito.

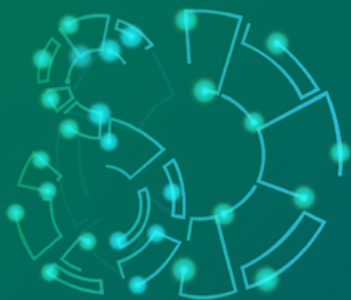




Para lembrar...

- Processos de software são “complexos”, tanto que, muitas vezes, estamos falando de “modelos de processos” e não processos propriamente ditos. Modelos são abstrações que apresentam elementos essenciais e evitam os detalhes.
- Vimos que comunicação, planejamento, modelagem, construção e implantação fazem parte da espinha dorsal do desenvolvimento de software.¹¹ Essas atividades, sejam para criar um software, como parte da evolução de um software ou até mesmo a correção de um bug são esforços realizados no escopo de um projeto de software.
- Temos dois ciclos de vida pertinentes a software, um de produto (software) e outro de projeto. Enquanto o ciclo de vida de um software tem como foco o software, o ciclo de vida de um projeto de software tem como foco o gerenciamento de tal projeto.
- Projeto de software envolve questões como a distribuição do esforço ao longo do tempo, financiamento, férias e outros.





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Engenharia de software na saúde digital

Unidade 3 Elaboração de Proposta de Software Integrado à Rede Nacional de Dados em Saúde

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior



Unidade 3: Elaboração de Proposta de Software Integrado à Rede Nacional de Dados em Saúde

Nas Unidades anteriores foram apresentados vários tópicos. A intenção foi fornecer fundamentos que trazem certo grau de compreensão do que é realizado ao longo da produção de *software*.

Nesta Unidade, vamos orientá-lo(a) a contribuir com a definição de um SIS com ênfase no componente tecnológico (*software*) desse sistema. A elaboração de uma proposta é uma atividade que você irá desenvolver. Aqui, nos limitaremos a orientá-lo(a) por essa jornada.

É necessário ser um engenheiro de *software* para “definir” um *software*? Definir pode ser interpretado de forma rigorosa e, nesse caso, só a formação de um profissional poderia produzir o resultado de uma atividade de “análise do negócio” ou “análise de requisitos”. Observe que nem mencionamos essas atividades anteriormente, não é verdade? Por outro lado, quem conhece de um domínio, tal como um profissional de saúde quando o *software* é no domínio da saúde, está em uma posição privilegiada para orientar a atribuição de funções a um *software*. Adicionalmente, quem está nessa posição privilegiada e está empoderado(a) com tecnologias oferecidas pelo País, como a RNDS, torna-se um(a) profissional ainda mais importante na identificação de oportunidades.

Nesse sentido, a seção seguinte apresenta um roteiro para orientar o leitor a contribuir com a definição de uma oportunidade que, em uma sequência lógica natural, seria avaliada na perspectiva econômica, dentre outras considerações posteriores. O roteiro minimalista produz insumos que alimentam tais considerações. Trata-se de um passo inicial razoável para quem está em uma posição privilegiada.

Antes do roteiro, o termo oportunidade foi empregado acima, também podemos acrescentar outros como transformação digital e inovação. A intenção é nos preparar para contribuir com o que está na pauta presente, a saber, a inovação baseada em *software*, conforme Mani e Ebert (2022).²

3.1 Roteiro Minimalista

Não temos a pretensão de definir um método no sentido literal do termo, mas de fornecer orientações. Nesse caso, as orientações serão baseadas em algumas atividades fundamentais, a saber:

- registrar sua ideia de forma clara;
- verificar aderência da ideia aos objetivos de negócio;
- descrever processos pertinentes; e
- identificar requisitos do *software*.

Essa proposta deve ser elaborada em grupos de 2 a 3 discentes.



3.1.1 Sua Ideia e Riscos

Você conhece a saúde, conhece seu ambiente de trabalho e teve uma ideia de como “melhorar” a qualidade da assistência oferecida. Talvez pela redução do tempo da assistência, talvez pela simplificação de processos executados pelos profissionais de saúde, ou pela simplificação na perspectiva dos indivíduos assistidos, enfim, você sabe como “melhorar”.

Essa melhoria está associada à transformação digital do contexto em questão. Ou seja, sua ideia inclui um sistema de informação ou a modificação de um existente, o que significa que um novo serviço a ser executado por um *software* estaria disponível. Vamos simplificar aqui e supor que o *hardware* no qual é executado é um computador convencional ou um *smartphone* ou está disponível nas “nuvens”.



Fonte: <https://storyset.com/>

Sua ideia pode ser disruptiva, revolucionária. Nesse caso, talvez existam muitos riscos envolvidos, o que provavelmente significa que terá que desenvolver protótipos, não é verdade? Afinal, você tem uma ideia que vem acompanhada de incertezas. Talvez você esteja bem seguro da sua ideia, mas outros podem demandar protótipos ou uma prova de conceito, ou um projeto piloto, pelo simples fato de não compartilhar a mesma segurança que você possui.

Sua ideia pode ser “genial” no sentido em que é “fácil” perceber que é exequível e, adicionalmente, “funciona”, introduz melhoria e de baixo custo. Nesse caso, sua ideia é realmente “genial”.

Naturalmente, entre os extremos acima existem vários cenários. Talvez você seja uma pessoa avessa a riscos, talvez essa aversão não limite suas ideias criativas. De qualquer forma, considerando o mundo de rápidas mudanças em que vivemos, há risco em incertezas e também há riscos em permanecer onde se está.

Mas, vamos concluir com uma citação para te inspirar:

“Alguns homens vêem as coisas como são e perguntam por quê.
Outros sonham coisas que nunca existiram e perguntam por que não.”

Bernard Shaw

3.1.2 Qual a Melhoria Introduzida pela Sua Ideia?

Agora que já tem uma ideia, é bom que ela resolva alguma dificuldade e, dessa forma, introduza melhoria. É bem provável que tenha iniciado pela dificuldade e buscado uma solução.

É preciso confrontar sua ideia com os objetivos do negócio (aqui são os objetivos da sua unidade de saúde). Afinal, ninguém deseja a introdução de um novo



Fonte: <https://storyset.com/>



computador apenas pelo fato dele ser mais veloz, mas, provavelmente, porque irá reduzir o tempo gasto com algum processamento necessário.

Objetivo de negócio é o que se espera atingir, obter. Por exemplo, reduzir o tempo médio desde a chegada de um indivíduo a um estabelecimento até o instante em que é assistido por um profissional de saúde.

Sempre há um custo que se deseja reduzir ou a resolutividade de alguma demanda significativa de uma região, e assim por diante.

Por que identificar objetivos de negócio é importante? Porque toda ação de um ator de suporte como a Tecnologia da Informação (TI), por exemplo, só faz sentido se estiver alinhada a tais objetivos. Também, deixa claro que a transformação digital não é um objetivo em si, mas um caminho para se atingir um objetivo do negócio em questão. Você também estará apto, uma vez que objetivos de negócio são conhecidos, a descartar eventuais propostas que não contribuem com tais objetivos.

Observe, por exemplo, a possibilidade de emprego de realidade aumentada por meio do [HoloLens](#)²⁴ em ambientes de telessaúde. Seguramente, você pode imaginar várias utilidades e recursos. Contudo, apesar da possibilidade de transformação digital proporcionada por tal dispositivo, talvez essa opção não esteja alinhada ao contexto de alguns estabelecimentos de saúde, enquanto possa ser para outros. A transformação digital (sua ideia) está a serviço dos objetivos de negócio e não o contrário.

Por fim, sua ideia não precisa fazer uso de tecnologias “sofisticadas” como aquela citada acima, sua ideia precisa ajudar a resolver uma questão do domínio da saúde.

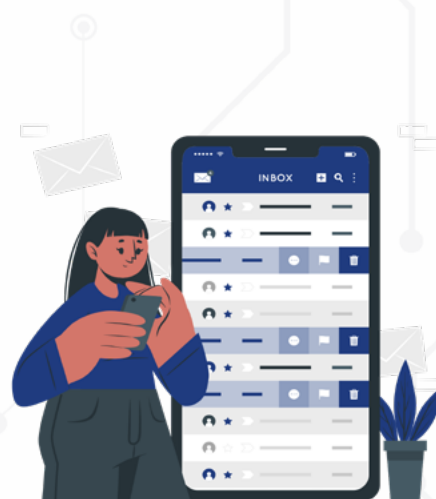
3.1.3 Processos

Você já tem uma ideia, já verificou que gera valor para o estabelecimento de saúde em questão (alinhamento com os objetivos de negócio) e, não necessariamente nessa ordem, precisa mostrar como funciona a sua ideia. Em geral, uma ideia envolve outros componentes de um sistema de informação, novamente, tarefas, pessoas e papéis, além da tecnologia (*software* comentado no item seguinte).

Uma alternativa para mostrar esse funcionamento é documentar os processos de negócio pertinentes na BPMN,¹³ dentre outras opções.

3.1.4 Requisitos

Na presença de uma ideia que se mostra alinhada aos objetivos de negócio de um estabelecimento de saúde e dos processos envolvidos, conforme as seções anteriores, pode-se identificar as funções que ficarão a cargo do *software*, afinal, estamos no contexto de transformação digital da saúde. A definição dos requisitos é particularmente importante para atribuir ao *software* um papel bem definido no escopo da sua ideia.



Fonte: <https://storyset.com/>



Estamos particularmente interessados, no nosso contexto, em requisitos de usuário. Um requisito de usuário descreve uma tarefa que usuários desejam realizar com um *software* visando a um resultado de valor. Observe que, com tal definição, nada melhor que o próprio usuário para contribuir com a identificação de tais requisitos.

Convém observar que existem várias outras formas de se registrar requisitos de *software*. Também existem processos específicos acerca de como identificá-los, além de outras questões pertinentes cujos detalhes podem ser obtidos em referências específicas.²⁵ Isso não nos impede, contudo, enquanto “não especialistas em requisitos”, de oferecer nossa contribuição. A expectativa é de que esse processo possa produzir um conteúdo adequado para análise posterior de especialistas em *software*.

Uma forma de registrar requisito de usuário é por meio do que é conhecido por *user story*. O formato padrão é:

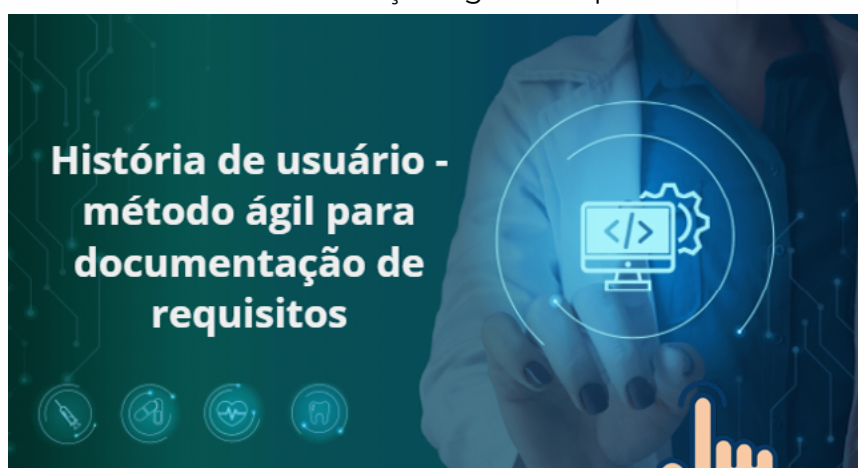
Como um <tipo de usuário>, eu quero <um objetivo> para <alguma finalidade>

Alguns exemplos apenas para ilustrar o formato acima, sem escopo (domínio) específico:

- Como enfermeiro, eu quero consultar a prescrição de medicamentos a serem administrados a um dado paciente para dar continuidade ao cuidado.
- Como membro de uma equipe de Estratégia de Saúde da Família, eu desejo enviar uma notificação a um dos pacientes atendidos para fornecer informação relevante para a continuidade do cuidado.
- Como gestor da telessaúde do município, eu desejo reservar a sala de telessaúde para assistência remota.
- Como gestor do município, eu desejo consultar a fila de espera por cirurgia eletiva para orientar as pactuações.
- Como usuário assistido, eu desejo consultar a data, horário, local e outras informações pertinentes à consulta agendada para dar continuidade ao meu tratamento.

Assista, a seguir, a Videoaula 3 que trata sobre *user stories* (histórias de usuário), um modelo de especificação de requisitos que, de forma simples, define e organiza as necessidades do usuário sob o ponto de vista dele mesmo^{18,19} ministrada pela Prof^a. Dra. Renata Dutra Braga.

Videoaula 3 - Documentação ágil de requisitos de software



Fonte: autoria própria.



LINK

3.2 Considerações sobre Sua Proposta

Nesse esforço de definição de uma proposta de serviço ou produto para a saúde, a ser realizado, provavelmente por profissional de saúde, identificamos quatro atividades:

- a. a identificação clara da ideia;
- b. a verificação do alinhamento da ideia aos objetivos do estabelecimento de saúde em questão (ou outro escopo, como o município, por exemplo);
- c. apresentação do funcionamento da ideia por meio dos processos introduzidos e/ou alterados; e
- d. a atribuição de um papel para o *software* que é parte da ideia.

Ainda, convém destacar um forte aliado a ser utilizado pela sua proposta: a RNDS. Nem toda transformação digital em saúde no País necessariamente envolve a RNDS, contudo, nesta Unidade estamos particularmente interessados naquelas que envolvem a RNDS. Ou seja, conte com a possibilidade de informação em saúde trafegar entre estabelecimentos de saúde por meio da RNDS.

Sabemos que um engenheiro de *software* tem à disposição todo um conhecimento específico, além de ferramentas e processos, para realizar essas atividades, contudo, outros profissionais que conhecem o domínio em questão, são de participação imprescindível nesse processo. Observe que um engenheiro de *software* funciona como um “habilitador tecnológico” nesse processo de transformação digital, capitaneado pela saúde.

No domínio da saúde, é natural esperar que profissionais da saúde, conhecedores das dificuldades da prática que exercem, contribuam efetivamente com a identificação da transformação digital. São esses profissionais que podem “desenhar” processos de assistência diferentes daqueles em uso, reorganizar um determinado serviço e contar com o apoio de um *software* para o qual eles vão definir as funcionalidades.

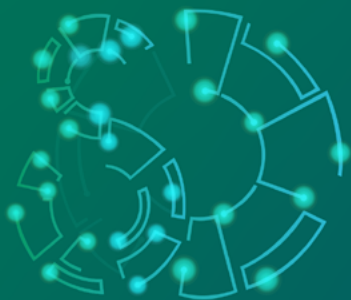




Para lembrar...

- Identifique com precisão o domínio que será afetado positivamente pela transformação digital. Por exemplo: recebimento de laudos de exames onde solicitante e executante estão geograficamente separados.
- Descreva sua ideia, talvez em um único parágrafo. Por exemplo: atualmente laudos são fisicamente transportados, uma vez por semana, de vários laboratórios até a nossa unidade de saúde. A ideia é substituir esse processo manual, oneroso, moroso e propenso a perdas, por um sistema no qual profissionais de saúde dos laboratórios, respectivamente munidos de seus certificados digitais, assinam digitalmente laudos por eles produzidos e, na sequência, eletronicamente disponibilizados para a nossa unidade de saúde, o que elimina o transporte, os custos e a demora correspondentes, além de ampliar a segurança.
- Mostre que sua ideia está alinhada aos objetivos do seu estabelecimento de saúde. Por exemplo: agilizar a assistência ao paciente é melhorar a qualidade do serviço de saúde oferecido, principalmente pela assistência realizada o quanto antes possível. Além da agilidade, os custos operacionais da solução serão reduzidos, o que mitiga os custos de desenvolvimento ao longo do tempo.
- Seguramente, sua ideia introduz mudanças no contexto em questão. Essas mudanças interferem em papéis, atividades e outros. Com o propósito de esclarecer essas mudanças ou o funcionamento da sua ideia, é bom que ofereça uma forma de compreendê-la. Um instrumento sugerido é o registro explícito de processos de negócio, possivelmente, por meio do emprego da BPMN,¹³ por exemplo.
- Se os processos de negócio foram caracterizados, então, foi atribuído um papel (funcionalidades) para um software que, dentre outras, interage com a RNDS. Essa caracterização no nível de processos é muito abstrata e precisa ser refinada. O instrumento sugerido para tal é o registro de requisitos de usuário. Por exemplo: o sistema ou software existente terá que ser acrescido de funcionalidade que permita o recebimento eletrônico de laudos. Duas funcionalidades são identificadas abaixo:
 - Como responsável pelo recebimento de laudos, eu desejo anexar cada laudo recebido, após minha verificação, ao prontuário eletrônico do paciente em questão, para que a continuidade do cuidado possa considerá-lo no momento oportuno.
 - Como responsável pelo recebimento de laudos, eu desejo consultar os laudos recebidos por nome do paciente, para verificar a ausência de laudo esperado, mas ainda não recebido.





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Engenharia de software na saúde digital

Unidade 4 **Análise, Revisão e Apresentação das Propostas**

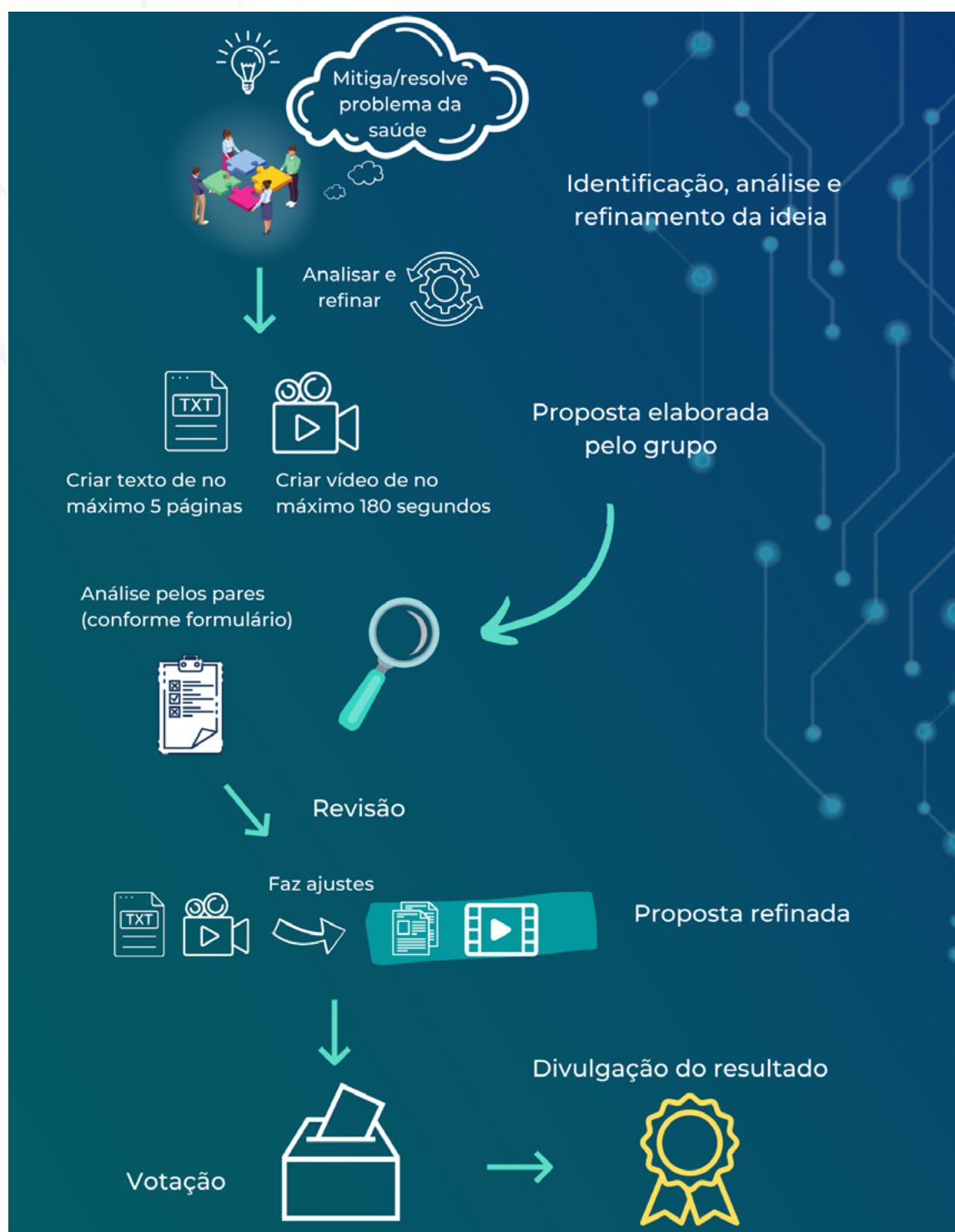
Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior



Unidade 4: Análise, Revisão e Apresentação das Propostas

Na Unidade anterior, você foi orientado(a) acerca da elaboração de uma proposta de *software* integrado à RNDS. Nesse ponto, você já elaborou essa proposta, ou melhor, o seu grupo já tem uma proposta devidamente produzida. Doravante, são fornecidos outros detalhes acerca do registro da proposta e outros. Todo o processo está ilustrado na Figura 9.

Figura 9 - Processo para a elaboração de uma proposta de *software* integrado à Rede Nacional



Fonte: autoria própria.

4.1 Registro da Proposta

Sua proposta será registrada por dois instrumentos:

- a. documento no formato PDF: no máximo, cinco páginas, papel A4 e fonte de tamanho 12 pt; e
- b. vídeo: no máximo, três minutos.

Cinco páginas não é muito. Pouco menos de 200 segundos também não. Contudo, cada um deles deve revelar ao seu consumidor uma ideia de transformação digital. Faça-os de tal forma que possam ser “consumidos” independentemente. Ou seja, o vídeo não é para explicar o documento.

A ideia, além das orientações já fornecidas, provavelmente tem um nome, deixa claro o valor criado e a relação custo/benefício, dentre outros. É possível que, no documento textual, você inclua um diagrama registrando processos de negócio, por exemplo, enquanto talvez não seja tão adequado para ser exibido em um vídeo curto.

De qualquer forma, essas são apenas dicas gerais para que uma “grande ideia” não seja minimizada ou corra o risco de nem ser percebida por conta de um formato inadequado. Isso não significa que é preciso contratar um serviço de editoração para o documento, nem tampouco uma produtora de vídeo. O documento provavelmente fará uso de ferramentas como Google Document®, Microsoft Office® Word®, Canva® ou similares. O vídeo será filmado com o seu celular, *webcam* ou por um *software* como *screencastify*,²⁶ dentre uma variedade de opções gratuitas, diga-se de passagem.

É bom lembrar que, na plateia, há um ser humano, além das orientações abaixo para avaliação da proposta.

Cada proposta deve ser elaborada em grupo de 2 a 3 discentes. Apenas com a devida justificativa e aprovação da Coordenação da Especialização, será admitido um grupo com um número inferior de participantes.

4.2 Análise pelos Pares

Cada proposta será avaliada exatamente por um grupo. Nenhum grupo avaliará a sua própria proposta. A distribuição de qual grupo irá avaliar qual proposta ficará a cargo da Coordenação.

A avaliação de uma proposta deve resultar em sugestões de melhorias, tanto no texto quanto no vídeo. Uma melhoria é um ajuste na ideia para torná-la ainda mais atrativa ou corrigir uma falha. Uma melhoria também pode ter como alvo os instrumentos (o documento e o vídeo). O texto pode conter algum erro, alguma imagem do vídeo pode ser substituída ou algo do gênero.

A avaliação será registrada em formulário a ser disponibilizado posteriormente. Cada grupo irá preencher o formulário conforme o conteúdo recebido para avaliação.

Cada formulário preenchido (avaliação) será entregue ao grupo da proposta avaliada, conforme a seção seguinte.



4.3 Revisão

Cada proposta avaliada (formulário preenchido) será encaminhada aos autores (grupo que produziu a proposta). O objetivo é que cada grupo tenha a oportunidade de rever a sua proposta e, dessa forma, melhorá-la, à luz da avaliação feita. Cabe a cada grupo fazer ajustes ou não na sua proposta. Noutras palavras, a revisão oferece a oportunidade, não a obrigatoriedade, para que cada grupo altere o seu documento e/ou o vídeo. Seja a ideia alterada ou não, por meio do documento e do vídeo, será encaminhada para a próxima etapa.

4.4 Eleição

Provavelmente, até esse momento, cada estudante conhece apenas duas propostas, a sua e aquela avaliada. Nessa etapa, todas as propostas (documento e vídeo correspondentes) serão disponibilizadas para todos, estudantes e professores(as).

Cada estudante e cada professor(a) votará individualmente em três propostas, nessa ordem, primeiro, segundo e terceiro lugares. O estudante não poderá incluir a sua proposta em nenhuma das três votadas. Ou seja, cada estudante só poderá votar nas propostas dos demais colegas.

O resultado será uma amostra do que valorizamos e, seguramente, um instrumento de aprendizado para todos.

4.5 Divulgação do resultado

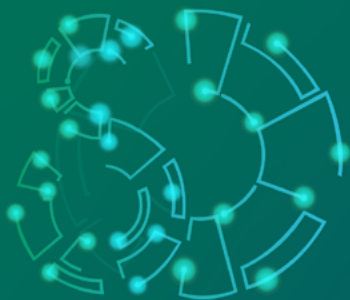
O resultado será divulgado com a identificação das três primeiras propostas mais votadas. A intenção, convém deixar claro, é identificar as propostas que refletem, dentre aquelas apresentadas, o que consideramos mais relevante.



Para lembrar...

- A definição de uma proposta de transformação digital não é uma competição. Muitas ideias, provavelmente, serão colocadas em prática tendo em vista as necessidades da saúde.
- O esforço realizado durante esse processo é um esforço que será, provavelmente, realizado por cada estudante em seu contexto profissional.





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Engenharia de software na saúde digital

Unidade 5 **Encerramento do Microcurso**

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior



Unidade 5: Encerramento do Microcurso

A Engenharia de *Software* é uma área do conhecimento relativamente recente, quando comparada com a milenar química, por exemplo. O termo *software* é da década de 50, e apenas no final da década de 60 foi cunhada a expressão Engenharia de *Software*. De qualquer forma, hoje, já existe um vasto conhecimento acerca da construção de programas de computador registrado nessa área.

Impulsionado pelo que um *software* proporciona, a sociedade tornou-se uma ávida consumidora e dependente de *software*. Na saúde não é diferente. A saúde humana, contudo, ainda não usufrui de benefícios que podem ser obtidos por meio de sistemas de informação e que já são usufruídos por outras áreas. Por exemplo, o registro de vacinas aplicado no gado bovino já é realizado pelo Ministério da Agricultura, Pecuária e Abastecimento (MAPA) empregando o Sistema Brasileiro de Identificação e Certificação de Bovinos e Bubalinos (SISBOV), desde 2011. Numa época em que muitas transações financeiras são realizadas via PIX, é natural esperar que informações em saúde também possam trafegar com a mesma agilidade e segurança, talvez até superior, visando à assistência a cada cidadão.

Nesse contexto, profissionais de saúde são atores que estão em posição privilegiada para definir, contribuir com o desenvolvimento, apresentar oportunidades e melhorias. E agora, após serem expostos a um pouco de como a Engenharia de *Software* funciona, a comunicação com os profissionais da Engenharia de *Software* é apoiada, o que deve repercutir positivamente na saúde.

Vamos avançar para o último próximo Microcurso da Área Temática Inovações e Tendências: “**Transformação digital na saúde**”? Nos encontramos lá!



Referências

1. WIKIPEDIA. **Domain (software engineering)**. Acesso em 20 jan. 2022. Disponível em: [https://en.wikipedia.org/wiki/Domain_\(software_engineering\)](https://en.wikipedia.org/wiki/Domain_(software_engineering)).
2. MANI, V. S.; EBERT, C. Medical software. **IEEE Software**. 2022, v. 39, n. 1, p. 11-18. Acesso em 21 jan. 2022. Disponível em: <https://ieeexplore.ieee.org/document/9662367>.
3. WIKIPEDIA. **Information system**. Acesso em 20 jan. 2022. Disponível em: https://en.wikipedia.org/wiki/Information_system.
4. INFOQ. **About InfoQ: accelerating the software side of human progress**. Acesso em 20 jan. 2022. Disponível em: <https://www.infoq.com/about-infoq/>.
5. SOFTWARE ENGINEERING RADIO. **About SE-Radio**. Acesso em 20 jan. 2022. Disponível em: <https://www.se-radio.net/about/>.
6. INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS (IEEE). **Software Engineering Body of Knowledge (SWEBOK)**. Acesso em 20 jan. 2022. Disponível em: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>.
7. INVESTOPEDIA. **The 10 Richest People in the World**. Acesso em 20 jan. 2022. Disponível em: <https://www.investopedia.com/articles/investing/012715/5-richest-people-world.asp>.
8. WIKIPEDIA. **List of public corporations by market capitalization**. Acesso em 20 jan. 2022. Disponível em: https://en.wikipedia.org/wiki/List_of_public_corporations_by_market_capitalization.
9. CURCIO, K. NAVARRO, T.; MALUCELLI, A.; REINEHR, S. Requirements engineering: a systematic mapping study in agile software development. **Journal of Systems and Software**. 2018, v. 139, p. 32-50. Acesso em 20 jan. 2022. Disponível em: <https://doi.org/10.1016/j.jss.2018.01.036>.
10. WIKIPEDIA. **System**. Acesso em 20 jan. 2022. Disponível em: <https://en.wikipedia.org/wiki/System>.
11. PRESSMAN, R. S.; MAXIM, B. R. **Software engineering: a practitioner's approach**. 9. ed. New York: McGraw-Hill, 2019. 704 pp.



12. QUOTE INVESTIGATOR. **Plans are worthless, but planning is everything.** Acesso em 20 jan. 2022. Disponível em: <https://quoteinvestigator.com/2017/11/18/planning/>.
13. WIKIPEDIA. **Business Process Model and Notation.** Acesso em 20 jan. 2022. Disponível em: https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation.
14. WIKIPEDIA. **List of programming languages.** Acesso em 20 jan. 2022. Disponível em: https://en.wikipedia.org/wiki/List_of_programming_languages.
15. TIOBE. **TIOBE Index for April 2022.** Acesso em 20 jan. 2022. Disponível em: <https://www.tiobe.com/tiobe-index/>.
16. WIKIPEDIA. **Petri net.** Acesso em 20 jan. 2022. Disponível em: https://en.wikipedia.org/wiki/Petri_net.
17. PROJECT MANAGEMENT INSTITUTE; AGILE ALLIANCE. **Agile practice guide.** Project Management Institute, 2017. 241 pp.
18. CHEN, I.. **Agile Business Intelligence (BI) development methodology and best practices** [Internet]. Jan, 2017. Acesso em 20 abr. 2022. Disponível em: <https://chen115yaohua.wordpress.com/2017/01/10/40/>.
19. KANE, D. W.; HOHMAN, M. M.; CERAMI, E. G.; MCCORMICK, M. W.; KUHLMMAN, K. F.; BYRD, J. A.. Agile methods in biomedical software development: a multi-site experience report. **BMC Bioinformatics**. 2006, v. 7, n. 273, p. 1-12. Acesso em 20 abr. 2022. Disponível em: <https://doi.org/10.1186/1471-2105-7-273>.
20. WIKIPEDIA. **Sutyagin house.** Acesso em 20 jan. 2022. Disponível em: https://en.wikipedia.org/wiki/Sutyagin_House.
21. BREDENOORD, J.. **Incremental housing.** Acesso em 20 jan. 2022. Disponível em: <https://bredenoordhousingresearch.com/incremental-housing/>.
22. WIKIPEDIA. **Iterative and incremental development.** Acesso em 20 jan. 2022. Disponível em: https://en.wikipedia.org/wiki/Iterative_and_incremental_development.
23. THE INTERNATIONAL ORGANIZATION FOR STANDARDIZATION; THE INTERNATIONAL ELECTROTECHNICAL COMMISSION. **ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.** 2011. Acesso em 21 jan. 2022. Disponível em: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>.



24. MICROSOFT. **HoloLens2**. Acesso em 20 jan. 2022. Disponível em: <https://www.microsoft.com/en-us/hololens/>.
25. WIEGERS, K.; BEATTY, J. **Software requirements**. 3. ed. Washington: Microsoft Press. 2013. Acesso em 21 jan. 2022. Disponível em: http://wdz.eng.br/WDS/assets/files/Software_Requirements_3rd_Edition.pdf.
26. SCREENCASTIFY. **Screencastify: video for everyone**. Acesso em 21 abr. 2022. Disponível em: <https://www.screencastify.com/>.



Minibiografias

Organizadores

Fábio Nogueira de Lucena é graduado em Ciência da Computação (UFG), mestre e doutor em Ciência da Computação (UNICAMP), especialista em Informática em Saúde (UNIFESP), Project Management Professional (PMI) e Certified Software Development Professional (IEEE), além de possuir outras certificações da indústria de software. É professor titular do curso de Engenharia de Software do Instituto de Informática da UFG.

Github: <https://github.com/kyriosdata>

E-mail: kyriosdata@ufg.br

Ana Laura de Sene Amâncio Zara é graduada em Farmácia e em Análises Clínicas (UFMT), especialista em Avaliação de Tecnologias em Saúde (UFRGS) e em Docência do Ensino Superior (UCDB). Possui mestrado e doutorado em Epidemiologia pelo Programa de Pós-Graduação em Medicina Tropical e Saúde Pública (UFG) e pós-doutorado pelo Programa de Pós-graduação de Odontologia da Faculdade de Odontologia (UFG). Atualmente, é professora do Departamento de Saúde Coletiva da UFG. Ensina, pesquisa e orienta nas áreas de Epidemiologia, Saúde Coletiva, Metodologia e Editoração Científicas, Economia da Saúde, Bioestatística, Informática em Saúde e Revisões Sistemáticas.

E-mail: analaufg@gmail.com

Plínio de Sá Leitão Júnior é Engenheiro Eletricista, com mestrado e doutorado pela Universidade Estadual de Campinas (UNICAMP) na área de Engenharia de Software, e Especialização em Informática em Saúde pela Universidade Federal de São Paulo (UNIFESP). É Professor Associado no Instituto de Informática da Universidade Federal de Goiás (UFG), com atuação na graduação e na pós-graduação. Desenvolve pesquisas nos temas Teste de Software, Banco de Dados, Inteligência Computacional e Persistência de Registros Clínicos.

E-mail: plinio.sa.leitao.junior@ufg.br

Rejane Faria Ribeiro-Rotta é graduada em Odontologia (UFG), especialista em Radiologia Bucomaxilofacial e Estomatologia, mestre e doutora em Odontologia (Diagnóstico Bucal) (USP-Bauru), com experiência em colaborações internacionais em pesquisa e intercâmbios, e na gestão institucional do ensino superior. Professora titular da Faculdade de Odontologia da UFG. Fundadora do Centro Goiano de Doenças da Boca da Faculdade de Odontologia da UFG (CGDB-FO-UFG) e da Comissão de Governança da Informação em Saúde da UFG. Principais temáticas de pesquisa: Diagnóstico de lesões da região bucomaxilofacial / Câncer de boca; Dores crônicas orofaciais; Diagnóstico por imagem da região bucomaxilofacial; Prática baseada em evidência, Informação e Informática em saúde.

E-mail: rejanefrr@ufg.br

Renata Dutra Braga é professora adjunta do Instituto de Informática da Universidade Federal de Goiás (UFG). É mestre e doutora em Ciências da Saúde pela Faculdade de Medicina da UFG, pós-graduada em Informática em Saúde (UNIFESP) e em Qualidade e Gestão de Software (PUC-GO) e é graduada em Sistemas de Informação (UniEvangélica). É atualmente vice-coordenadora da Comissão de Governança da Informação em Saúde (CGIS-UFG). Ensina, pesquisa, orienta e desenvolve projetos de extensão na área de saúde digital, com interesse, principalmente em modelagem de processos de negócios, engenharia de requisitos, modelos de informação, terminologias clínicas e padrões para a troca da informação em saúde.

E-mail: renatadbraga@ufg.br

Rita Goreti Amaral é professora titular da Faculdade de Farmácia da Universidade Federal de Goiás (UFG), com atuação na graduação e pós-graduação. Graduada em Farmácia e Bioquímica e especialista em Citologia Clínica (UFG). Mestre em Biologia Celular e Molecular (USP) e Doutora em tocoginecologia pela Faculdade de Ciências Médicas (UNICAMP). Coordenadora do Laboratório de Monitoramento Externa da Qualidade da Faculdade de Farmácia (UFG). Desenvolve projetos de pesquisa e extensão na área de Citologia Clínica e Saúde Pública, atuando nos seguintes temas: controle da qualidade em citopatologia do colo do útero, prevenção, detecção precoce de doenças, aperfeiçoamento de métodos diagnósticos, desenvolvimento e validação de práticas de cuidado do paciente nas doenças crônicas transmissíveis e não transmissíveis, informática em saúde e assistência farmacêutica.

E-mail: rita@ufg.br

Sheila Mara Pedrosa é graduada e mestre em Enfermagem pela Faculdade de Enfermagem (UFG), especialista em Saúde Coletiva e Regulação em Saúde no SUS (IEP/HSL) e doutora em Ciências da Saúde pela Faculdade de Medicina (UFG). Atualmente é professora adjunta do Centro Universitário de Anápolis e desenvolve pesquisa e extensão no âmbito das violências e vulnerabilidade social. É membro da Comissão de Governança da Informação em Saúde (CGIS-UFG) e participa de projetos voltados à saúde digital.

E-mail: sheilaenf@gmail.com

Silvana de Lima Vieira dos Santos - é enfermeira, mestre e doutora em Ciências da Saúde (UFG), Especialista em Enfermagem em Infectologia (USP) e em Informática em Saúde (UNIFESP). É professora associada da Faculdade de Enfermagem (UFG). Vice líder do Núcleo de Estudos e Pesquisa de Enfermagem em Prevenção e Controle de Infecção Relacionada à Assistência à Saúde (NEPIH), vinculado ao CNPq. Experiência na área de prevenção e controle de infecções relacionadas à assistência à saúde, epidemiologia e informática em saúde. Coordenadora da Comissão de Governança da Informação em Saúde (CGIS-UFG).

E-mail: silvanalvsantos@ufg.br

Taciana Novo Kudo é professora adjunta do Instituto de Informática da Universidade Federal de Goiás (UFG). É mestre e doutora em Ciência da Computação pelo Departamento de Computação (UFSCar) e graduada em Ciência da Computação (UNIMAR). Possui experiência profissional na área de Engenharia de Software, especificamente em Engenharia de Requisitos e Gerência de Projetos, em institutos de pesquisa e empresas de São Paulo e Goiás. Como pesquisadora, atua em projetos voltados para Engenharia de Software, Engenharia de Requisitos e Informática aplicada à Educação e à Saúde.

E-mail: taciana@ufg.br



PROGRAMA
EDUCACIONAL
EM **SAÚDE
DIGITAL**
DA UNIVERSIDADE
FEDERAL DE GOIÁS



SOBRE O E-BOOK

Tipografia: Montserrat

Publicação: Cegraf UFG

Câmpus Samambaia, Goiânia -
Goiás. Brasil. CEP 74690-900

Fone: (62) 3521-1358

<https://cegraf.ufg.br>