

Adriano Luiz de Souza Lima

**UsabilidadeZero:
Um aplicativo para ensino de usabilidade**

Florianópolis, SC

2 de julho de 2019

Adriano Luiz de Souza Lima

**UsabilidadeZero:
Um aplicativo para ensino de usabilidade**

Proposta de trabalho de conclusão de curso a ser apresentada como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação pela Universidade Federal de Santa Catarina – UFSC

Universidade Federal de Santa Catarina

Orientadora: Fabiane Barreto Vavassori Benitti

Florianópolis, SC
2 de julho de 2019

Resumo

Interfaces de software com boa usabilidade ajudam o usuário a completar mais tarefas, em menor tempo e com menos esforço, proporcionando-lhe maior satisfação. Em alguns casos, a usabilidade da interface pode representar o sucesso ou o fracasso comercial de um sistema de software frente à grande quantidade de opções disponíveis para os usuários hoje em dia. Apesar da sua importância na área de Tecnologia da Informação, foi percebido que poucas ferramentas de apoio didático ao tema estão disponíveis para professores e alunos da área. Saber como medir a usabilidade de interfaces quando utilizadas para alcançar objetivos específicos é um dos conceitos básicos que se deve apreender quando se inicia no estudo do tema. O presente trabalho tem como objetivo principal apresentar o desenvolvimento de uma aplicação web que possa ser utilizada por professores como recurso didático de apoio para o ensino de usabilidade a alunos de graduação. Ao utilizar a aplicação, alunos da disciplina Engenharia de Software do curso de graduação em Sistemas de Informação que nunca tiveram contato com o tema puderam identificar os principais aspectos da usabilidade e descobrir como eles podem ser medidos em uma interação para realizar tarefas específicas. Além disso, eles puderam reconhecer alguns critérios de usabilidade e perceber como as medidas são alteradas quando alguns desses critérios são adequadamente aplicados ou não no design de interfaces. Como resultado, mais de 90% desses alunos declararam terem percebido a importância da usabilidade para a qualidade de um produto de software e mais de 80% afirmaram que a experiência com o aplicativo vai contribuir para seu desempenho na vida profissional.

Palavras-chave: Engenharia de Software, Interação Humano-computador, Usabilidade, Ferramentas de Ensino

Lista de tabelas

Tabela 1 – Tabela Comparativa dos Critérios de Usabilidade	20
Tabela 2 – Dados Extraídos de cada Estudo Primário Selecionado	24
Tabela 3 – Resultado da Busca	25
Tabela 4 – Resumo dos Artigos Selecionados	27
Tabela 5 – Identificação dos Artigos Selecionados	28
Tabela 6 – Classificação de Tipo de Pesquisa	30

Lista de ilustrações

Figura 1 – Processo de Seleção dos Estudos Primários	25
Figura 2 – Casos de Uso para a Aplicação UsabilidadeZero	35
Figura 3 – Diagrama de atividades detalhando o caso de uso UC1.2	36
Figura 4 – Casos de Uso para a Aplicação Portal de Eventos	36
Figura 5 – Modelo Entidade-Relacionamento	38
Figura 6 – Tela inicial	40
Figura 7 – Tela de instruções para o estudo de caso	41
Figura 8 – Tela de relatório após o término da tarefa	42
Figura 9 – Tela do formulário SUS (visão parcial)	42
Figura 10 – Tela de relatório após o término do estudo de caso	43
Figura 11 – Tela de relatório final	44
Figura 12 – Tela de inicial do Portal de Eventos no EC0	45
Figura 13 – Tela de inicial do Portal de Eventos no EC1	45
Figura 14 – Tela de detalhes de evento no EC0	46
Figura 15 – Tela de detalhes de evento no EC1	47
Figura 16 – Formulário de inscrição em evento no EC0 (visão parcial)	49
Figura 17 – Formulário de inscrição em evento no EC1 (visão parcial)	50
Figura 18 – Formulário de emissão de certificado no EC0	51
Figura 19 – Formulário de emissão de certificado no EC1	52
Figura 20 – Estrutura hierárquica do modelo GQM	53
Figura 21 – Quantidade de cliques por usuário em cada um dos estudos de caso	58
Figura 22 – Tempo por usuário em cada um dos estudos de caso	58
Figura 23 – Eficácia por usuário em cada um dos estudos de caso	59
Figura 24 – Satisfação por usuário em cada um dos estudos de caso	60
Figura 25 – Reconhecimento das medidas de usabilidade antes e depois do uso do aplicativo	61
Figura 26 – Reconhecimento de como medir a usabilidade antes e depois do uso do aplicativo	62
Figura 27 – Reconhecimento de critérios de usabilidade	63
Figura 28 – Métricas M2.4.1, M2.4.2 e M2.4.3	64
Figura 29 – Medidas de usabilidade escolhidas antes e depois do uso do aplicativo	66
Figura 30 – Critérios de usabilidade reconhecidos	67

Sumário

	Lista de tabelas	3
	Lista de ilustrações	4
1	INTRODUÇÃO	7
1.1	Problema de Pesquisa	8
1.2	Solução Proposta	8
1.3	Justificativa	9
1.4	Objetivos	9
1.4.1	Objetivo Geral	10
1.4.2	Objetivos Específicos	10
1.5	Delimitação do Escopo	10
1.6	Metodologia	11
1.6.1	Método de Pesquisa	11
1.6.2	Procedimentos Metodológicos	11
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Interação Humano-Computador	13
2.2	Usabilidade	13
2.2.1	Princípios, Regras e Orientações	14
2.2.1.1	As Regras de Ouro de Schneiderman	14
2.2.1.2	As Heurísticas de Usabilidade de Nielsen	15
2.2.1.3	Os Critérios Ergonômicos de Bastien e Scapin	16
2.2.1.4	Os Princípios do Diálogo da ISO 9241	19
2.2.2	Análise Comparativa	19
2.2.3	Medidas de Usabilidade	21
3	MAPEAMENTO SISTEMÁTICO	22
3.1	Planejamento e Condução do Mapeamento	22
3.2	Resultados	24
4	DESENVOLVIMENTO	31
4.1	Estudos de Caso	31
4.2	Especificação dos Requisitos	32

4.2.1	Aplicação UsabilidadeZero	32
4.2.1.1	Requisitos Funcionais	32
4.2.1.2	Regras de Negócio	33
4.2.2	Portal de Eventos	34
4.2.2.1	Requisitos Funcionais	34
4.2.2.2	Regras de Negócio	34
4.2.3	Requisitos Não Funcionais	34
4.3	Casos de Uso	35
4.3.1	UsabilidadeZero	35
4.3.2	Portal de Eventos	36
4.4	Aplicação	37
4.4.1	Modelagem dos Dados	37
4.4.2	Camada Cliente	37
4.4.3	Camada Servidor	38
4.5	Operacionalidade	39
4.5.1	UsabilidadeZero	39
4.5.2	Portal de Eventos	41
5	AVALIAÇÃO	53
5.1	Planejamento	53
5.2	Execução	57
5.3	Resultados	57
5.4	Discussão	64
6	CONSIDERAÇÕES FINAIS	69
6.1	Trabalhos Futuros	69
	REFERÊNCIAS	71
	APÊNDICE A – FORMULÁRIO DE AVALIAÇÃO	77
	APÊNDICE B – MAPEAMENTO DOS CRITÉRIOS DE USABILIDADE	86
	APÊNDICE C – ARTIGO	89
	APÊNDICE D – CÓDIGO-FONTE	105

1 Introdução

Interfaces com boa usabilidade melhoram a interação do usuário com o dispositivo, proporcionando-lhe eficácia, eficiência e satisfação. Ao contrário, quando o usuário não entende como a interface funciona, então o produto de software torna-se inútil para ele (POLACK-WAHL, 2004), o que torna o desenvolvimento de uma boa interface tão importante quanto o desenvolvimento das outras partes do sistema.

Como uma das vantagens de uma interface com boa usabilidade está o bem estar do usuário (CYBIS; BETIOL; FAUST, 2010). Ao utilizar uma boa interface, o usuário sente-se satisfeito e confiante, pois consegue realizar o seu trabalho com qualidade e num tempo satisfatório. Quando, ao contrário, a interface é ruim, o usuário tende a se sentir frustrado pela diminuição da sua produtividade e o excesso de erros cometidos.

Na web, devido à grande quantidade de páginas disponíveis, há grande competitividade pela atenção do usuário. Por esse motivo, ao menor sinal de dificuldade para operar uma página online, os usuários a abandonam e vão para outra (NIELSEN, 2012a). Essa atitude costuma se refletir nas vendas e na imagem da empresa no mercado.

Já em ambientes de trabalho fechados, onde o usuário não pode simplesmente largar o sistema e utilizar outro, como faria com uma página de internet, baixa usabilidade geralmente se traduz em baixa produtividade (NIELSEN, 2012a). Cada minuto a mais que o usuário perde tentando entender como realizar uma ação no sistema ou em treinamento para aprender a utilizá-lo é dinheiro perdido pela empresa.

Como consequência, a preocupação com a usabilidade tende a ser indispensável para que as empresas possam alcançar o sucesso comercial dos seus sistemas. Usuários satisfeitos indicam o sistema ou o serviço para outras pessoas que podem vir a se tornar novos clientes.

O ensino de usabilidade é conteúdo importante de engenharia de software, fazendo parte de diretrizes nacionais e internacionais para o ensino de computação, como o currículo de referência para graduação em ciência de computação e engenharia de software, da Sociedade Brasileira de Computação (SBC, 2005), e as diretrizes para currículos de graduação em engenharia da computação, elaborado em conjunto pela *Association for Computing Machinery* (ACM) e a pela *IEEE Computing Society* (ACM/IEEE-CS Joint Task Force on Computing Curricula, 2013). Apesar de sua importância, ferramentas específicas que auxiliem o seu ensino são quase inexistentes (SOMMARIVA, 2012). O presente trabalho pretende desenvolver uma ferramenta que possa ser usada como apoio didático a professores do tema.

1.1 Problema de Pesquisa

Nos últimos anos o desenvolvimento da tecnologia mudou consideravelmente a forma como o ser humano se comunica. Os telefones celulares, por exemplo, possuíam pouca utilidade além de fazerem chamadas telefônicas. Poucas pessoas tinham acesso à internet e a comunicação por esse canal se dava, basicamente, por email. Nos dias de hoje a comunicação é feita por troca de mensagens de texto, de áudio, de imagem, de vídeo e de símbolos, de forma síncrona ou assíncrona. Paralelamente, a educação sempre esteve de portas abertas para o uso de tecnologias, como o uso de simuladores, para citar apenas um exemplo. Apesar disso, a comunicação entre professores e alunos dentro da sala de aula de uma universidade parece não ter mudado da mesma forma. Aulas expositivas nas quais os professores apresentam o conteúdo da matéria aos alunos que, se tiverem alguma dúvida, fazem suas perguntas aos professores ainda são a maioria (DJAJALAKSANA, 2011). Esse tipo de aula tem muita dificuldade de atrair o interesse dos alunos (WANGENHEIM; WANGENHEIM, 2012) que, como consequência, apresentam baixa taxa de retenção do conteúdo (BRABRAND, 2010).

Outra característica dos métodos tradicionais de ensino é o pouco envolvimento dos alunos na transmissão do conteúdo, uma vez que, em geral, esse métodos favorecem uma postura passiva frente aos professores, que têm que ser muito mais ativos em suas aulas (BRABRAND, 2010). Como esses métodos oferecem poucas oportunidades aos alunos de aplicar os novos conhecimentos em novas situações concretas, tem-se, como consequência, um nível de aprendizagem superficial, que não ultrapassa as categorias iniciais do domínio cognitivo da taxonomia de Bloom (FERRAZ; BELHOT, 2010), por exemplo.

O ensino de usabilidade nos cursos da área de computação não tem sido muito diferente. A multidisciplinaridade do tema nem sempre tem sido levada em conta pelos professores que têm optado por ensiná-la por meio da análise ou projeto de uma simples interface (LESTER, 2008). Existem alguns cursos de usabilidade, por exemplo (RUSU; RUSU, 2006) que parecem, no entanto, indicar o caminho para um ensino mais completo. Faltam, entretanto, ferramentas específicas para o ensino do tema e que poderiam auxiliar o professor nesse processo (SOMMARIVA, 2012).

1.2 Solução Proposta

A partir da constatação da escassez de ferramentas que sirvam de suporte pedagógico para o ensino de usabilidade (SOMMARIVA, 2012), este trabalho propõe o desenvolvimento de uma aplicação web que possa servir como recurso didático no ensino do tema, apresentando a alunos de graduação os principais aspectos de usabilidade, conforme definição da ISO 9241:11 (ISO, 2018), e a importância da observação de critérios de usabilidade no desenvolvimento de uma interface.

Ao utilizar a aplicação, o aluno poderá interagir com interfaces que tiveram sua usabilidade

propositalmente reduzida e também com interfaces que observem critérios que aumentem a sua usabilidade. Espera-se que com essa experiência ele possa conhecer os principais aspectos de usabilidade e como medi-los. O aluno deve poder também descrever os critérios faltantes ou utilizados em cada um dos casos, mesmo que para isso ele não utilize uma terminologia precisa. Ao final de sua experiência, deseja-se que o aluno possa entender a importância da usabilidade para a qualidade de um produto de software.

A abordagem proposta pelo uso da ferramenta para o ensino de usabilidade aproxima-se da *aprendizagem baseada em problemas*, uma forma de metodologia ativa caracterizada pelo uso de problemas do mundo real que possam encorajar os alunos a desenvolverem seu pensamento crítico sobre o tema ao aplicarem seus conhecimentos e habilidades para resolvê-lo (BORGES; ALENCAR, 2014).

1.3 Justificativa

Não obstante a dificuldade de se encontrar ferramentas tecnológicas específicas para o ensino de usabilidade, algumas iniciativas vêm sendo apresentadas, principalmente na forma de jogos (SOMMARIVA, 2012; FERREIRA et al., 2014; BATTISTELLA, 2016). Ocorre, no entanto, que outras ferramentas parecem ser praticamente inexistentes. Essas ferramentas podem tornar o conteúdo das aulas mais prático, mostrando a relação existente entre os conceitos teóricos as aplicações reais ou simuladas. Além de tornar o aprendizado mais atrativo, ferramentas didáticas podem oferecer uma estratégia instrucional diferente das bastante utilizadas aulas expositivas, desde que sejam flexíveis o suficiente para que possam se adequar aos objetivos pedagógicos dos professores.

A solução proposta neste trabalho pretende oferecer aos alunos a oportunidade de lidarem com um problema concreto e tentarem resolvê-lo, assemelhando-se com a aprendizagem baseada em problemas (BORGES; ALENCAR, 2014). Ao utilizar a ferramenta, o professor pode levar os alunos a usar conhecimentos prévios em novas situações, a subdividir o conteúdo em partes menores para entender a estrutura final, ou juntar partes para criar um novo todo, proporcionando-lhes a oportunidade de atingir as categorias mais elevadas de aplicação, análise e síntese no domínio cognitivo da Taxonomia de Bloom (FERRAZ; BELHOT, 2010).

1.4 Objetivos

Os objetivos geral e específicos são apresentados como se segue:

1.4.1 Objetivo Geral

O objetivo deste trabalho é desenvolver uma aplicação web que possa ser utilizada em sala de aula como recurso didático de apoio para o ensino de aspectos e critérios de usabilidade a alunos de graduação.

1.4.2 Objetivos Específicos

Este trabalho possui os seguintes objetivos específicos:

- a) Estudar aspectos de usabilidade propostas pelos principais autores, identificando aqueles que são mais comumente mencionados;
- b) Proceder mapeamento sistemático sobre as metodologias usadas no ensino de IHC, com especial interesse em usabilidade, e a utilização de ferramentas tecnológicas nesse processo;
- c) Desenvolver uma aplicação web na qual estudantes possam realizar três tarefas em dois ambientes distintos, sendo o primeiro deles um ambiente cuja interface tenha sua usabilidade reduzida e o segundo um ambiente cuja interface atenda aos critérios de usabilidade identificados no item acima;

1.5 Delimitação do Escopo

Este trabalho busca o desenvolvimento de uma ferramenta que possa auxiliar o ensino de usabilidade. Essa ferramenta deverá ser bastante flexível para que o professor possa adequar o seu uso a seus objetivos. Por essa razão, não se pretende aqui oferecer um método para o ensino de usabilidade e nem mostrar o caminho que deve ser seguido em aula.

Além disso, não se pretende fazer um estudo mais aprofundado a respeito das diversas abordagens pedagógicas que poderiam ser utilizadas para o ensino de usabilidade. Pretende-se apenas disponibilizar a ferramenta, deixando a cargo dos profissionais do ensino atribuir-lhe o melhor uso.

Os aspectos de usabilidade abordados pela aplicação proposta são baseados na literatura sobre o tema e não há, neste trabalho, a pretensão de se trazer uma lista exaustiva ou definitiva a seu respeito.

Por fim, o conceito central estudado aqui é o de usabilidade, levando-se em conta seu ensino no nível de graduação. Isso quer dizer que outros aspectos da IHC, como psicologia, ciência cognitiva, ergonomia, fatores humanos e sociologia (LESTER, 2008) estão fora do escopo deste trabalho.

1.6 Metodologia

A seguir são apresentados o método de pesquisa desenvolvido durante o estudo, assim como os procedimentos metodológicos para levá-lo a cabo.

1.6.1 Método de Pesquisa

O presente trabalho classifica-se, quanto à sua natureza, como uma pesquisa aplicada, uma vez que objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos (SILVA; MENEZES, 2001) que, no presente caso, envolvem o conhecimento necessário para o desenvolvimento de uma ferramenta tecnológica para o ensino de usabilidade.

Esta pesquisa aborda o problema de modo qualitativo, considerando que não se pretende traduzir em números os dados de pesquisa e suas relações, mas interpretá-los e atribuir-lhes significados de forma descritiva (SILVA; MENEZES, 2001).

Por fim, quanto aos seus objetivos, a presente pesquisa é exploratória, uma vez que visa aproximar-se e familiarizar-se com o problema de modo a torná-lo explícito ou a construir hipóteses (GIL, 2008, p.27). Para isso, será necessária a condução de um levantamento bibliográfico acerca do tema.

1.6.2 Procedimentos Metodológicos

O desenvolvimento deste projeto deverá ocorrer em concordância com o seguinte plano de trabalho:

Etapa 1 – Fundamentação teórica: trata-se do estudo e revisão dos conceitos básicos necessários para o desenvolvimento do projeto. As atividades que compõem esta etapa são as seguintes:

- a) IHC e usabilidade: estudo teórico dos conceitos principais a serem explorados durante o trabalho, suas definições e importância;
- b) Critérios de usabilidade: levantamento e estudo dos principais critérios de usabilidade utilizadas para avaliação de interfaces;
- c) Análise comparativa entre os critérios de usabilidade: verificação das semelhanças entre os principais conjuntos de critérios de usabilidades propostos.

Etapa 2 – Introdução do projeto: etapa inicial do projeto de pesquisa, com a definição do problema e planejamento das etapas futuras:

- a) Problema de pesquisa: definição do problema de pesquisa e proposta de solução, justificativa, objetivos (geral e específicos), delimitação do escopo e metodologia;

- b) Proposta de TCC: elaboração da proposta de TCC a ser apresentada como requisito para a disciplina *Introdução a Projetos*.

Etapa 3 – Mapeamento sistemático: pesquisa e revisão de estudos e trabalhos correlatos na área e suas possíveis contribuições. Contém as seguintes atividades (KITCHENHAM; CHARTERS, 2007):

- a) Planejamento: definição das perguntas de pesquisa e produção do protocolo de revisão;
- b) Execução: condução da revisão a partir do protocolo definido;
- c) Resultados: escrita do relatório com os resultados da revisão.

Etapa 4 – Implementação da aplicação: etapa mais longa de todo o projeto, na qual o sistema é implementado. Suas atividades são as seguintes:

- a) Análise e projeto: levantamento dos requisitos necessários para o desenvolvimento do sistema e sua modelagem, com a confecção do Documento de especificação de requisitos;
- b) Desenvolvimento dos módulos da aplicação: etapa que consumirá a maior parte do tempo do projeto com o desenvolvimento da aplicação.
- c) Testes de sistema: etapa de testes das funcionalidades da aplicação.

Etapa 5 – Avaliação dos resultados e conclusão: etapa na qual a avaliação é planejada e aplicada e seus resultados analisados para que se possa chegar a conclusões a respeito do trabalho realizado. Divide-se nas seguintes atividades:

- a) Planejamento da avaliação: definição da estratégia para aplicação da avaliação;
- b) Aplicação da avaliação: avaliação do sistema por parte dos usuários finais e sugestões de melhorias;
- c) Análise dos resultados e conclusão: após a aplicação da avaliação, os resultados serão analisados sob a luz da teoria para que possam conduzir a conclusões adequadas.

2 Fundamentação Teórica

2.1 Interação Humano-Computador

É notório que o computador tem ganhado importância cada vez mais central na vida dos seres humanos. Suas aplicações, que antes eram usadas apenas para atender a necessidades profissionais, hoje possibilitam que milhões de usuários ao redor do mundo possam se comunicar e colaborar de diversas maneiras (SHNEIDERMAN; PLAISANT, 2009, p.4). Essa mudança na forma como o ser humano lida com a máquina se deve ao desenvolvimento das interfaces cada vez mais fáceis de usar.

Inicialmente, os computadores eram operados diretamente via hardware, ou seja, eram programados por operadores que conectavam suas válvulas e circuitos por meio de cabos e chaves e recebiam o resultado da computação programada em mostradores ou impresso (PREECE; ROGERS; SHARP, 2005, p.28). Essa forma de operar o computador exigia profissionais bastante treinados, geralmente engenheiros. Com a evolução, os usuários começaram a ver os dados de entrada e de saída de uma computação em interfaces que eram exibidas em telas de monitores, que passaram a ser o espaço de interação entre o usuário e o computador. Foi a partir da necessidade de entender esse tipo de interação que se desenvolveu a disciplina chamada *Interação Humano-Computador* (IHC), uma área de estudo multidisciplinar que abarca áreas do conhecimento como a psicologia, o design, a ciência da computação, a semiótica, entre outras.

Um dos primeiros desafios enfrentados pela IHC foi desenvolver interfaces que possibilitassem a diversos tipos de profissionais diferentes, além dos engenheiros, utilizar os computadores para exercerem atividades que envolvessem a cognição humana (PREECE; ROGERS; SHARP, 2005, p.30). Posteriormente, com o desenvolvimento de novas tecnologias e com a popularização dos computadores pessoais e, mais tarde, dos dispositivos móveis, as interfaces deveriam se adequar às necessidades de praticamente qualquer pessoa. Portanto, é preciso que as interfaces fossem fáceis de usar para que, mesmo sem treinamento, os usuários pudessem operá-la para realizar suas tarefas. Essa é a qualidade de interfaces com usabilidade.

2.2 Usabilidade

Usabilidade diz respeito ao grau de facilidade com que um usuário consegue utilizar uma ferramenta para realizar uma certa tarefa (PREECE; ROGERS; SHARP, 2005; NIELSEN, 2012a).

Diversos autores apresentam suas próprias definições de usabilidade. Conforme apresentado em Cybis, Betiol e Faust (2010, p.16), usabilidade “depende de um acordo entre as características de sua interface e as características de seus usuários ao buscarem determinados objetivos em

determinadas situações de uso.”

Para Nielsen (2012a), “usabilidade é um atributo de qualidade que avalia como as interfaces de usuário são fáceis de usar.” Além disso, usabilidade é definida por cinco componentes de qualidade: a) capacidade de aprender; b) eficiência; c) capacidade de memorizar; d) erros; e e) satisfação.

Preece, Rogers e Sharp (2005, p.35) ensinam que “usabilidade é geralmente considerada como o fator que assegura que os produtos são fáceis de usar, eficiente e agradáveis – da perspectiva do usuário.”

No presente trabalho, é usada a definição de usabilidade da ISO 9241:11 que a tem como a medida pela qual um sistema, produto ou serviço pode ser usado por usuários específicos para alcançar objetivos específicos com efetividade, eficiência e satisfação em um contexto de uso específico (ISO, 2018).

2.2.1 Princípios, Regras e Orientações

Para analisar se uma dada interface possui ou não problemas de usabilidade, pode-se verificar se ela segue alguns princípios ou critérios de usabilidade. Esse é um método de baixo custo de avaliação, também conhecido como avaliação heurística. Entre os critérios mais populares estão as *Regras de Ouro*, de Schneiderman, as *Heurísticas de Usabilidade*, de Nielsen, e os *Critérios Ergonômicos*, de Bastien e Scapin.

2.2.1.1 As Regras de Ouro de Schneiderman

As “Regras de Ouro” foram originalmente apresentadas em (SHNEIDERMAN; PLAISANT, 2009) e atualizadas em (SHNEIDERMAN, 2016).

1. **Esforçar-se pela consistência:** Ações consistentes devem ser mantidas em situações similares, com terminologia idêntica, cores, leiaute, fontes, etc., consistentes por toda a interface, apesar de ser aceitável a aplicação de exceções como, por exemplo, a substituição de senhas por asteriscos;
2. **Buscar a usabilidade universal:** O reconhecimento das necessidades dos diferentes usuários, facilitando a transformação do conteúdo, enriquece o design da interface e aumenta a qualidade percebida;
3. **Oferecer feedback informativo:** Cada ação executada pelo usuário deve receber um feedback na proporção de sua importância; ações frequentes e de menor impacto devem receber uma resposta modesta, enquanto que ações mais significativas devem receber respostas mais substanciais;

4. **Desenhar diálogos que indiquem o fim de uma ação:** Sequências de ações devem ser organizadas de forma que tenham começo, meio e fim e recebem um feedback informativo na sua conclusão; assim o usuário terá satisfação pelo fim da execução e poderá se preparar para o próximo grupo de ações;
5. **Prevenir erros:** Sempre que possível a interface deve evitar que os usuários cometam erros (por exemplo, protegendo campos de formulário inapropriados ou não permitindo a entrada de caracteres alfabéticos em campos numéricos) e, caso eles ocorram, devem oferecer instruções simples, construtivas e específicas para a recuperação;
6. **Permitir fácil reversão das ações:** Sempre que possível a interface deve permitir a reversão das ações, aliviando a ansiedade dos usuários ao saber que seus erros poderão ser corrigidos e os encorajando a explorar opções desconhecidas;
7. **Manter usuários em controle:** Usuários, principalmente os mais experientes, querem ter total controle da interface e saber que não encontrarão surpresas no seu comportamento;
8. **Reduzir a carga de memória de curto prazo:** Como a capacidade humana de processar informação na memória de curto prazo é limitada, deve-se evitar que os usuários tenham que lembrar informações de uma tela para serem usadas em outra.

2.2.1.2 As Heurísticas de Usabilidade de Nielsen

As dez heurísticas a seguir foram apresentadas por Rolf Molich e Jakob Nielsen inicialmente em (MOLICH; NIELSEN, 1990). A versão atual foi refinada em (NIELSEN, 1994).

1. **Visibilidade do estado do sistema:** O sistema deve informar ao usuário o que está acontecendo, fornecendo o feedback adequado dentro de um espaço de tempo razoável;
2. **Mapeamento entre o sistema e o mundo real:** A linguagem falada pelo sistema deve ser familiar ao usuário, evitando termos específicos do sistema, seguindo convenções do mundo real de forma que a informação seja apresentada uma ordem natural e lógica;
3. **Liberdade e controle ao usuário:** O usuário deve ter a liberdade de escolher interromper facilmente uma tarefa do sistema quando perceber que escolheu uma função equivocada;
4. **Consistência e padrões:** Termos, situações e ações iguais devem sempre corresponder ao mesmo significado em toda a interface;
5. **Prevenção de erros:** Um sistema desenhado de forma a evitar que erros sejam cometidos pelo usuário é preferível do que tratar e corrigir esses erros;

6. **Reconhecer em vez de lembrar:** Os objetos, ações e opções devem estar visíveis e as instruções devem estar facilmente acessíveis ao usuário de forma a minimizar a carga de informação que deve ser lembrada;
7. **Flexibilidade e eficiência de uso:** O sistema deve oferecer facilidade de uso a usuários inexperientes e atalhos que possam acelerar o trabalho de usuários experientes;
8. **Design estético e minimalista:** A interface deve apresentar apenas a informação necessária para a realização das tarefas a que se propõe, deixando de fora informações irrelevantes ou desnecessárias que diminuem a visibilidade relativa dos itens da interface;
9. **Suporte para o usuário reconhecer, diagnosticar e recuperar erros:** O sistema deve apresentar as mensagens de erro na linguagem do usuário, indicar de forma precisa o problema e sugerir uma solução de forma construtiva;
10. **Ajuda e documentação:** Apesar de ser melhor construir o sistema para ser utilizado sem a necessidade de documentação de ajuda, essa documentação deve existir e estar facilmente acessível ao usuário, além de não ser muito longa.

2.2.1.3 Os Critérios Ergonômicos de Bastien e Scapin

Os critérios ergonômicos apresentadas por Bastien e Scapin (1993) (CYBIS; BETIOL; FAUST, 2010) seguem abaixo.

1. **Condução:** A interface deve aconselhar, orientar, informar e conduzir o usuário em sua interação com o sistema, oferecendo facilidade de uso aos usuários e favorecendo o aprendizado dos novatos; está subdividida em *convite*, *agrupamento/distinção entre itens*, *feedback imediato* e *legibilidade*:
 - 1.1. **Convite:** Uma boa interface conduz o usuário para realizar ações específicas, fornecendo informações a respeito do estado e do contexto onde ele se encontra, disponibilizando fácil ajuda e reduzindo os erros;
 - 1.2. **Agupamento/distinção de itens:** A organização visual da interface, tanto espacial, quanto de formato, é um dos fatores primários para que o usuário possa compreender a relação entre seus elementos; este critério está subdividido em *agrupamento/distinção por localização* e *agrupamento/distinção por formato*:
 - 1.2.1. **Agupamento/distinção de itens por localização:** A localização relativa dos elementos da interface indica ao usuário se eles pertencem ao mesmo grupamento e, portanto, à mesma classe, ou se diferem entre si;

- 1.2.2. **Agrupamento/distinção de itens por formato:** As similaridades ou as diferenças entre os elementos de uma interface devem ser destacadas ao usuário por meio de formas gráficas com tamanho, cor, estilo, etc.
 - 1.3. **Feedback imediato:** Todas as entradas feitas pelo usuário devem receber, no tempo adequado para a tarefa a ser realizada, algum tipo de resposta do sistema;
 - 1.4. **Legibilidade:** A facilidade de leitura da informação apresentada na interface está intimamente ligada às suas características lexicais (brilho, contraste entre as letras e o fundo, tamanho da fonte, espaçamento entre linhas, etc.) e deve levar em consideração a capacidade cognitiva e perceptiva dos usuários.
2. **Carga de trabalho:** A diminuição da carga cognitiva e perceptiva exigida do usuário deve ser tomada pela interface como uma forma de reduzir erros e aumentar a eficiência da tarefa; está subdividida em *brevidade e densidade informacional*:
 - 2.1. **Brevidade:** A interface deve apresentar carga de trabalho limitada para a leitura e entrada de dados, assim como a quantidade de passos necessários para a execução da tarefa para encurtar o tempo necessário para realizá-la e diminuir o risco de cometer erros; está subdividida em *concisão e ações mínimas*:
 - 2.1.1. **Concisão:** Como a capacidade de memória de curto prazo do usuário é limitada, a interface deve favorecer entradas curtas;
 - 2.1.2. **Ações mínimas:** Tarefas devem ter um número de ações limitado e, por consequência, sua complexidade reduzida para reduzir a carga de trabalho.
 - 2.2. **Densidade informacional:** A quantidade de informação apresentada pela interface deve ser reduzida ao necessário para a realização da tarefa, evitando confundir o usuário e diminuir seu desempenho.
 3. **Controle explícito:** O sistema terá maior aceitação pelo usuário se ele puder controlar suas entradas, limitando erros e ambiguidades; está subdividida em *ações explícitas do usuário e controle do usuário*:
 - 3.1. **Ações explícitas do usuário:** Menos erros são observados quando o sistema processa apenas ações explicitamente solicitadas pelo usuário, principalmente se forem ações longas, sequenciais e de tratamento demorado ou que tenham repercursões importantes;
 - 3.2. **Controle do usuário:** O usuário deve ter a liberdade de escolher quando começar, interromper, alterar ou continuar qualquer tarefa, que deve ser antecipada pelo sistema por meio de opções apropriadas, tornando-lhe mais previsível.

4. **Adaptabilidade:** Um sistema que se comporta em conformidade com o contexto, oferecendo maneiras diversas de realizar uma tarefa e dando ao usuário a liberdade de escolher a que mais lhe convier, adapta-se melhor às suas necessidades e preferências; está subdividida em *flexibilidade e experiência do usuário*:
 - 4.1. **Flexibilidade:** A interface que oferece ao usuário diferentes caminhos para atingir o objetivo dado adapta-se às suas necessidades particulares e aumenta as chances que esse usuário escolherá um caminho e o dominará durante o aprendizado;
 - 4.2. **Experiência do usuário:** O nível de experiência do usuário deve ser levado em consideração na interação com interface, que pode apresentar diálogos de instrução para os usuários inexperientes e atalhos aceleradores de tarefas para os usuários experiente, por exemplo.
5. **Gestão de erros:** A interface deve cuidar para prevenir ou reduzir os erros dos usuários e cuidar para se recuperar quando eles ocorrem, evitando interrupções de sistema e melhorando o desempenho; está subdividida em *proteção dos erros, qualidade das mensagens de erro e correção de erros*:
 - 5.1. **Proteção dos erros:** É preferível detectar erros e evitá-los (ou pedir confirmação da ação ao usuário) antes que eles sejam validados do que depois, quando precisarão ser corrigidos;
 - 5.2. **Qualidade das mensagens de erro:** Mensagens de erro de qualidade são relevantes, legíveis e específicas a respeito da natureza do erro, e promovem o aprendizado do usuário;
 - 5.3. **Correção dos erros:** Erros acontecem, mas, se forem corrigidos imediatamente, tornam-se menos perturbadores.
6. **Coerência:** Características de design devem ser mantidas em contextos similares e diferentes em contextos diferentes de forma a deixar o sistema mais previsível e facilitar o aprendizado, reduzindo o tempo de busca por informações e a quantidade de erros cometidos;
7. **Significado dos códigos e denominações:** Uma interface clara, que relaciona adequadamente a informação apresentada ou pedida com a sua referência no sistema, torna a identificação do usuário novato e a lembrança do usuário intermitente mais fácil, evitando operações inapropriadas;
8. **Compatibilidade:** A transferência de informação é mais rápida e eficiente quando a quantidade de informação que deve ser traduzida, interpretada ou buscada na documentação é

limitada em uma interface que traz um acordo entre as características do usuário, da tarefa e do ambiente.

2.2.1.4 Os Princípios do Diálogo da ISO 9241

1. **Adaptação à tarefa:** A interface deve dar suporte para que o usuário complete sua tarefa com eficiência e eficácia, oferecendo apenas informação relevante para a tarefa ou evitando que passos desnecessários sejam tomados;
2. **Autodescrição:** O sistema de oferecer informações e feedback ao usuário sempre que ele solicitar ou necessitar delas (feedback), de forma a tornar cada passo do diálogo compreensível e a minimizar o uso de manuais de ajuda;
3. **Controlabilidade:** O usuário deve ter total controle da direção e da velocidade de suas interações, incluindo a interrupção ou a retomada das ações, até que suas tarefas estejam completas;
4. **Conformidade às expectativas do usuário:** A interface deve corresponder ao conhecimento, educação e experiência dos usuários, além de se conformar com as convenções mais amplamente aceitas;
5. **Tolerância a erros:** Mesmo que o usuário cometa erros de entrada de dados, o resultado esperado deve poder ser alcançado com nenhuma ou pequenas ações corretivas;
6. **Facilidade de individualização:** O sistema deve ser construído de forma que ele possa se adaptar e se modificar de acordo com as habilidades ou necessidades dos usuários;
7. **Facilidade de aprendizagem:** O sistema deve guiar o usuário através de seus níveis de aprendizagem, diminuindo o tempo necessário para que possa aprendê-lo.

2.2.2 Análise Comparativa

Ao analisar os critérios de avaliação de usabilidade das interfaces propostos acima, percebe-se que há muito em comum entre os autores, embora haja critérios que sejam apresentados por apenas um ou dois deles. A Tabela 1 relaciona esses critérios por meio de seu significado, mesmo que sua nomenclatura possa causar certa confusão a princípio. Por exemplo, o critério *liberdade e controle ao usuário* de Nielsen diz respeito à liberdade que o usuário tem para desfazer (ou refazer) ações indesejadas e, portanto, está mais próximo de *permitir fácil reversão das ações* e de *correção dos erros* do que de *manter usuários em controle* e de *controle do usuário*, de Schneiderman e de Bastien & Scapin respectivamente.

Na tabela, percebe-se que há quatro critérios que são compartilhados por todos os autores e outros dois que são compartilhados por três de quatro fontes. São eles:

Shneiderman	Nielsen	Bastien & Scapin	ISO 9241
Prevenir erros; Permitir fácil reversão das ações	Prevenção de erros; Liberdade e controle ao usuário	Proteção dos erros; Correção dos erros	Tolerância a erros
Esforçar-se pela consistência	Consistência e padrões	Coerência	Conformidade às expectativas dos usuários
Oferecer feedback informativo	Visibilidade do estado do sistema	Feedback imediato	Autodescrição
Buscar usabilidade universal	Flexibilidade e eficiência de uso	Experiência do usuário; Flexibilidade	Facilidade de individualização
Reduzir a carga de memória de curto prazo	Reconhecer em vez de relembrar	Ações mínimas; Concisão	
Manter usuários em controle		Ações explícitas do usuário; Controle do usuário	Controlabilidade
	Suporte para usuário reconhecer, diagnosticar e recuperar erros	Qualidade das mensagens de erro	
	Design estético e minimalista	Densidade informacional	
	Mapeamento entre o sistema e o mundo real	Significado dos códigos e denominações	
	Ajuda e documentação		Facilidade de aprendizagem
Desenhar diálogos que indiquem o fim de uma ação		Convite	Adaptação à tarefa
		Compatibilidade	
		Agrupamento/distinção de itens	
		Legibilidade	

Tabela 1 – Tabela Comparativa dos Critérios de Usabilidade

- a) **Tratamento de erros (prevenção e correção):** Evitar que erros sejam cometidos ou, após sua detecção, corrigi-los é sempre levado em consideração na construção de uma interface; Erros são indesejados, diminuem a produtividade e frustram o usuário;
- b) **Consistência dos elementos da interface:** Quando o usuário interage com uma interface, ele naturalmente espera que objetos iguais se comportem de maneira igual;
- c) **Feedback ao usuário:** Para a grande maioria dos usuários o computador é uma máquina misteriosa; Saber o que está acontecendo ajuda o usuário a lidar com ela;
- d) **Adaptação ao usuário:** Os computadores e seus sistemas foram feitos para atender aos humanos. Por isso, interfaces devem se adaptar às diversas necessidades e habilidades de seus usuários, não o contrário;
- e) **Redução da carga de trabalho:** Como uma ferramenta, o computador deve auxiliar o usuário a realizar suas tarefas com eficiência, reduzindo sua carga de trabalho ao mínimo possível;
- f) **Controle do usuário:** Todo o controle do sistema deve estar nas mãos dos usuários para que ele possa usá-lo para atender aos seus objetivos.

2.2.3 Medidas de Usabilidade

Na literatura não há consenso sobre como medir a usabilidade de uma interface, mas características como eficácia, eficiência e satisfação do usuário são comumente indicados como aspectos importantes a serem verificados em interfaces que buscam alta usabilidade (ISO, 2018; NIELSEN, 2012a; PREECE; ROGERS; SHARP, 2005).

A eficácia pode ser definida em termos de taxa de sucesso do usuário, ou seja, pelo percentual de tarefas que o usuário completa corretamente (NIELSEN, 2001). Além da execução correta e completa da tarefa, sua execução parcial ou com algum desvio do objetivo original também podem ser levadas em consideração. Nesse caso, pode ser atribuído um percentual inferior a 100%, com o percentual dependendo do grau de desvio.

A eficiência é relação entre a quantidade de recursos empregados na tarefa e o resultado obtido. Quanto menos recursos forem empregados para se chegar ao mesmo resultado, mais eficiente é o sistema. A eficiência de um usuário ao operar uma interface é caracterizada pela quantidade de tempo necessária para o seu aprendizado, pela rapidez em executar suas tarefas e pela quantidade de erros cometidos (NIELSEN, 2012a; SHNEIDERMAN; PLAISANT, 2009). A eficiência proporcionada por uma interface pode ser medida pelo tempo gasto pelo usuário para completar uma tarefa e/ou pela quantidade de cliques necessários para que ele possa atingir seu objetivo.

A satisfação está relacionada com grau de conforto com que o usuário opera a interface e sua aceitação (ISO, 2018). É um critério inerentemente subjetivo e, por esse motivo, sua medição é feita por meio de aplicação de questionário padronizado apropriado (NIELSEN, 2012b). A utilização de questionários padronizados permite a comparação entre resultados de participantes distintos. Atualmente, o questionário mais amplamente utilizado para esse fim é o *System Usability System* (SUS), que traz dez questões para serem respondidas numa escala de valores de 1 (Discordo Totalmente) a 5 (Concordo Plenamente). Um resultado considerado aceitável no questionário SUS deve ser de no mínimo 55 pontos (BANGOR; KORTUM; MILLER, 2008).

Essas características, no entanto, só fazem sentido tendo os usuários como referência, e esses podem ter os mais diversos perfis (iniciantes, experientes, ocasionais, frequentes, etc.) e possuir as mais diversas habilidades, origens, motivações, personalidades, culturas e estilos de trabalho (SHNEIDERMAN; PLAISANT, 2009, p.22).

3 Mapeamento Sistemático

Neste capítulo é apresentado um mapeamento sistemático desenvolvido para que se possa ter uma ampla visão sobre as aulas de IHC, com interesse especial em usabilidade. Em geral, a execução de um mapeamento sistemático, em vez de uma revisão bibliográfica, é preferível para investigar uma área de pesquisa mais ampla ou quando pouca ou nenhuma evidência acerca do tema é conhecida (KITCHENHAM; CHARTERS, 2007). O interesse aqui é ter uma indicação da quantidade dessa evidência, classificando-a tanto para este estudo, quanto para estudos posteriores.

3.1 Planejamento e Condução do Mapeamento

O principal objetivo deste mapeamento é identificar como a IHC é ensinada a alunos de graduação. Para guiar esta análise, é importante saber quais são as metodologias de ensino mais usadas em sala de aula e quais ferramentas, se alguma, são usadas para apoiá-las. A partir daí, três questões de pesquisa são propostas:

RQ1: Quais são as principais metodologias usadas para ensinar IHC para alunos de graduação?

RQ2: Quais são as ferramentas tecnológicas disponíveis, desenvolvidas especificamente para apoiar o ensino de IHC?

RQ3: Como as metodologias ou tecnologias usadas para ensinar IHC têm sido avaliadas?

Os estudos primários foram buscados nas seguintes fontes de pesquisa:

- ACM Digital Library (dl.acm.org);
- IEEE Xplore Digital Library (ieeexplore.ieee.org);
- Google Scholar (scholar.google.com);
- International Conference on Applied Human Factors and Ergonomics e Conferências Afiliadas, AHFE 2015 (AHRAM; KARWOWSKI; SCHMORROW, 2015) e AHFE 2017 (ANDRE, 2018);
- HCI International 2014-2017 (Tecnologias de Aprendizado e Colaboração) (ZAPHIRIS; IOANNOU, 2014a; ZAPHIRIS; IOANNOU, 2014b; ZAPHIRIS; IOANNOU, 2015; ZAPHIRIS; IOANNOU, 2016; ZAPHIRIS; IOANNOU, 2017a; ZAPHIRIS; IOANNOU, 2017b);

- HCI International 2013 (Design, Experiência do Usuário, e Usabilidade: Experiência do Usuário em Saúde, Aprendizado, Jogos, Cultural e Intercultural (Parte II) (MARCUS, 2013); Interface Humana e Gestão da Informação: Informação e Interação para Aprendizagem, Cultura, Colaboração e Negócios (Parte III)) (YAMAMOTO, 2013);

A busca na IEEE Xplore e na ACM Digital Library foi limitada aos estudos que tinham sido submetido a revisão por pares, isto é, revistas científicas e anais de congressos, e que foram publicados no período entre 2012 e 2017. Para complementar a busca nessas bases de dados, foi utilizado o Google Scholar, uma vez que seu poderoso motor de busca poderia encontrar estudos que não haviam sido encontrados inicialmente nos seus websites originais. Aqueles papers que foram retornados nas buscas na IEEE Xplore e na ACM Digital Library e que foram também encontrados pelo motor de busca do Google Scholar foram considerados apenas uma vez.

Ao buscar por anais de conferências que não estavam indexados naquelas bases de dados, não foi possível o uso de motores de busca e, por isso, os artigos tiveram que ser verificados diretamente na *homepage* de cada conferência. Algumas conferências, no entanto, possuíam um número tão grande de publicações que esse processo teve que ser automatizado. Esse foi o caso para a AHFE e conferências afiliadas. Nas suas listas de anais, as palavras “*usability*”, “*teaching*” e “*education*” foram buscadas individualmente utilizando a função *Localizar* do navegador. Assim que algum estudo era encontrado, os outros critérios de inclusão também eram verificados.

Os dados foram extraídos de cada um dos artigos selecionados seguindo 15 itens de informação previamente definidos (Tabela 2).

Para encontrar os estudos nas bases de dados citadas acima, a seguinte string de busca foi usada:

(“teaching hci” OR “hci teaching” OR “hci education” OR “teaching usability” OR “usability teaching” OR “usability education”) AND undergraduate AND (software OR game OR simulation OR tool OR environment OR methodology)

A primeira parte da *string*, antes do primeiro AND, foi elaborada para selecionar aqueles artigos que este estudo buscava, isto é, artigos que tratavam de ensino de IHC ou, mais especificamente, ensino de usabilidade. A segunda parte foi usada para limitá-los ao nível de graduação e a última parte tinha a intenção de encontrar entre esses artigos aqueles que mencionavam explicitamente como o ensino foi conduzido, usando algum tipo de ferramenta (software, jogo, simulação) ou metodologia, ou ambos.

O presente estudo tem como objetivo encontrar artigos que discutiam ou apresentavam algum tipo de metodologia para o ensino de IHC no nível de graduação. Aqueles artigos que, além da graduação, tratavam do ensino em outros níveis, como cursos de pós-graduação, também foram

¹ Topi et al. (2010)

² Petersen, Vakkalanka e Kuzniarz (2015)

Grupo	Item de Informação
Grupo 1: Identificação da publicação	II1. ID da publicação II2. Título da publicação II3. Ano da publicação II4. Nome do autor II5. Fonte da publicação
Grupo 2: Contexto mencionado na publicação	II6. Contexto da pesquisa (disciplina/graduação) II7. Tópicos específicos ensinados II8. Tópicos de IHC ¹ II9. Metodologia aplicada II10. Justificação para uso da metodologia II11. Ferramenta tecnológica usada II12. Uso da ferramenta
Grupo 3: Avaliação descrita na publicação	II13. Classificação do tipo de pesquisa ² II14. Avaliação II15. Amostra

Tabela 2 – Dados Extraídos de cada Estudo Primário Selecionado

considerados relevantes e foram incluídos na busca, embora dados relacionados a esses níveis não tenham sido extraídos.

Os artigos que não tratavam de aulas de IHC na graduação foram excluídos. Os artigos que, embora tratassem do ensino de IHC na graduação, discutiam sobre outros aspectos do curso, como proposta de currículo, por exemplo, também foram excluídos.

3.2 Resultados

O mapeamento foi conduzido de dezembro de 2017 a fevereiro de 2018 com a execução do protocolo, e resultou na seleção de 15 artigos para a extração de dados. O número de artigos retornados, examinados e selecionados de cada uma das fontes utilizadas está resumido na Tabela 3. O processo de seleção está ilustrado na Figura 1.

A busca na ACM Digital Library resultou em 6 artigos, dos quais 2 atendiam aos critérios de seleção. A busca diretamente na IEEE Xplore Library forneceu 25 estudos, dos quais 4 foram selecionados. Ao usar o Google Scholar, foram encontrados outros 48 artigos que não haviam retornado no motor de busca da ACM. Desses artigos, 4 estudos foram selecionados. O motor de busca do Google Scholar também forneceu 3 outros artigos da IEEE Xplore que não haviam sido encontrados nesta base de dados, mas que não foram incluídos na seleção.

Além daqueles estudos da ACM e da IEEE Xplore, Google Scholar também retornou 302 arti-

Base de Dados	# Artigos Encontrados	# Artigos Examinados	# Artigos Selecionados
ACM	6	6	2
IEEE	24	24	4
Google Scholar	353	100	9
AHFE	938	938	0
HCI International	439	439	0
Total	1.760	1.507	15

Tabela 3 – Resultado da Busca

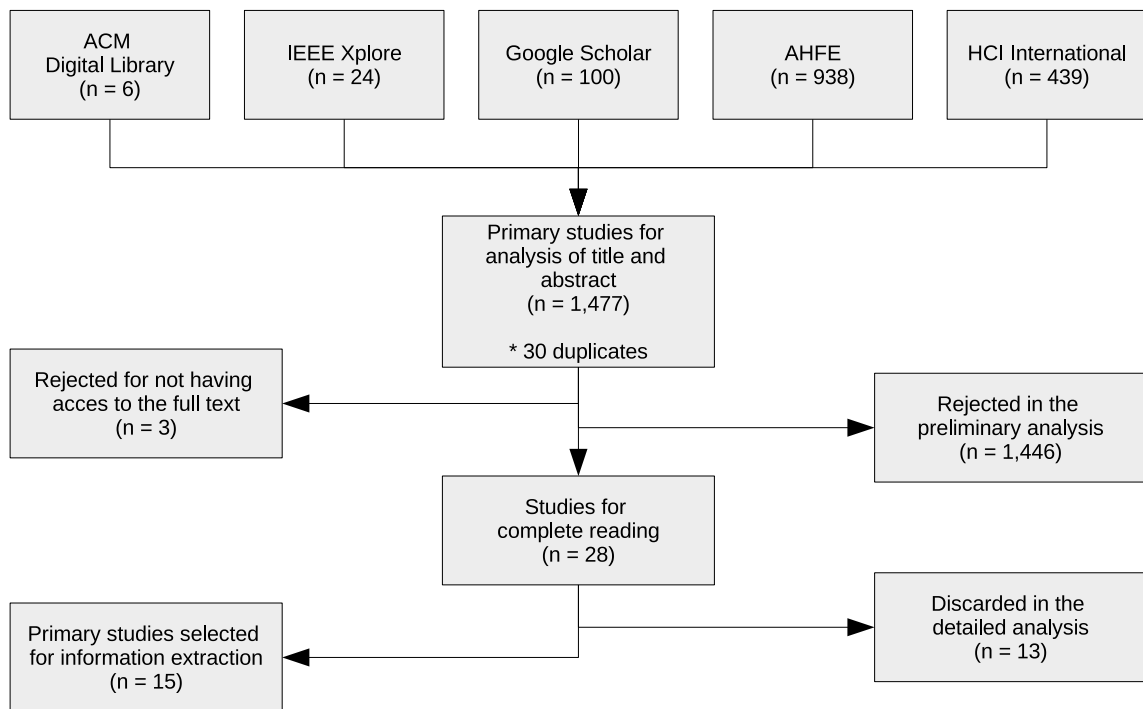


Figura 1 – Processo de Seleção dos Estudos Primários

gos de outras bases de dados, dos quais apenas os primeiros 100 estudos foram verificados, uma vez que aquele motor de busca ordena os artigos por relevância. Deste conjunto de artigos, 9 estudos atendiam aos critérios de seleção.

Na busca nos anais da International Conference on Applied Human Factors and Ergonomics (AHFE), apenas as conferências realizadas em 2015 e 2017 foram consideradas, uma vez que apenas nesses anos a conferência afiliada “Avanços nos Fatores Humanos em Treinamento, Educação e Ciências da Aprendizagem” aconteceu. A AHFE não foi realizada em 2013. A busca na conferência HCI International, em 2013, focou nas conferências afiliadas “Design, Experiência do Usuário, e Usabilidade: Experiência do Usuário em Saúde, Aprendizado, Jogos, Cultural e Intercultural (Parte II)” e “Interface Humana e Gestão da Informação: Informação e Interação para Aprendizagem, Cultura, Colaboração e Negócios (Parte III)”, enquanto que entre os anos 2014 e 2017 o foco foi na conferência afiliada “Tecnologias de Aprendizado e Colaboração”. Essa conferência não foi realizada em 2012. Nenhum estudo foi selecionado nos anais dessas conferências.

A Tabela 4 resume os dados extraídos de todos os 15 artigos selecionados e que correspondem aos itens de informação dos grupos 1 e 2 da Tabela 2. A coluna *Contexto* indica onde a pesquisa descrita no artigo foi conduzida, ou seja, em qual disciplina, com o nome do curso de graduação em parênteses. A coluna *Tópicos no Texto* apresenta aqueles tópicos de IHC que os autores relataram que tinham sido trabalhados com os alunos durante suas pesquisas. Esses tópicos foram correlacionados com os tópicos da disciplina “Introdução à Interação Humano-Computador” (TOPI et al., 2010) e o resultado foi usado para preencher a coluna seguinte, *Tópicos de IHC*. Na próxima coluna, *Metodologia*, estão as metodologias usadas para ensinar IHC e, finalmente, a última coluna, *Ferramentas*, indica se o autor mencionou que tipo de ferramenta foi usada durante o processo. A Tabela 5 apresenta o identificador de cada publicação.

A partir dos dados extraídos dos artigos selecionados foi possível responder às perguntas de pesquisa.

RQ1 Quais são as principais metodologias usadas para ensinar IHC para alunos de graduação?

Os dados extraídos mostram que uma grande variedade de metodologias de ensino ou técnicas puderam ser identificadas (Tabela 4, coluna “Metodologia”). No entanto, excluindo 4 artigos que não mencionavam ou não deixavam claro qual metodologia estava sendo usada, todos os outros mencionavam algum tipo de abordagem de aprendizado ativo, considerado aqui como “qualquer método instrucional que envolva os alunos no processo de aprendizagem” (PRINCE, 2004).

Yi et al. (2017) usa uma abordagem baseada em interesse “para abordar a questão que os alunos podem apresentar uma falta de interesse em pesquisas contínuas e inovadoras depois de completar nosso curso em IHC”.

Para Zaharias, Belk e Samaras (2012), o aprendizado baseado em problemas desenvolve “o aprendizado experiencial e social e chama por uma abordagem ativa, vez de passiva, à aprendi-

ID	Contexto	Tópicos no Texto	Tópicos de IHC ³	Metodologia	Ferramentas
#1	IHC (CC)	Ciclo de vida de usabilidade; Análise de requisitos; Prototipação; Avaliação heurística	Desenvolvimento; Métodos de avaliação	Jogos sérios	Jogo de computador (UsabilityGame)
#2	DS (CC)	Análise de tarefas	Tópicos especiais em IHC	-	-
#3	IHC (CC)	Projeto de design de interface de usuário	Desenvolvimento	Aprendizado baseado em estúdio	Equipamento simples de arte; Ferramenta de prototipação (WOZ Pro); HTML
#4	IHC (CC)	Processo de design centrado no usuário com ênfase no design de interface	Design centrado no usuário	Aprendizado baseado em problema	Plataforma de mundo virtual (OpenSimulator)
#5	IHC (CC, SI)	Teste de usabilidade	Métodos de avaliação	Aprendizado ativo	Ambiente de e-Learning centrado no aluno (SCELE)
#6	Curso de inverno (Eng)	Design de interação, usabilidade, experiência do usuário, prototipação, colecção de dados e análise	Design centrado no usuário; Desenvolvimento	Abordagem de pesquisa ativa	-
#7	IHC (SE)	Fundamentos básicos, Análise de requisitos, Design de interação, Avaliação de usabilidade	Princípios de design em IHC; Desenvolvimento; Métodos de avaliação	Desenvolvimento de projeto	-
#8	IHC (não informado)	Prototipação; Avaliação; Teste de usabilidade	Desenvolvimento; Métodos de avaliação	Modelo de aprendizado misto	sLearn
#9	CT (C, T, Eng, M)	Teste de usabilidade	Métodos de avaliação	-	-
#10	CT (Eng, CC, A)	Teste de usabilidade	Métodos de avaliação	-	-
#11	Aulas de IHC (P, D, Eng)	Ciclos centrais de design de IHC	Desenvolvimento	Abordagem em camadas	-
#12	IHC (CC, CI)	Princípios de design e usabilidade, com ênfase no lado humano das interações	Design centrado no usuário	-	-
#13	IHC (SGI)	Design de interface com ênfase nos usuários	Design centrado no usuário	Aprendizado baseado em estúdio	-
#14	IHC (CC)	Teorias cognitivas modais simples e múltiplas	Princípios de design em IHC	Aprendizado baseado em interesse	-
#15	IHC (CC)	Colecção e análise de requisitos; Design de interface; Avaliação de interface	Desenvolvimento; Métodos de avaliação	Abordagem de projetos realistas	-

CC: Ciência da Computação
SI: Sistemas de Informação
Eng: Engenharia
ES: Engenharia de Software
C: Ciências
T: Tecnologia

M: Matemática
A: Administração
P: Psicologia
D: Design
CI: Ciência da Informação
SGI: Sistemas de Gerenciamento de Informação

IHC: Interação Humano-Computador
DS: Design de Sistemas
CT: Comunicação Técnica

Tabela 4 – Resumo dos Artigos Seleccionados

ID	Referência	ID	Referência
#1	Benitti e Sommariva (2015)	#9	Chong (2017)
#2	Oliveira et al. (2015)	#10	Chong (2012)
#3	Hundhausen, Fairbrother e Petre (2012)	#11	Jeon (2015)
#4	Zaharias, Belk e Samaras (2012)	#12	Scialdone e Connolly (2016)
#5	Santoso e Sari (2015)	#13	Or-Bach (2015)
#6	Lazem (2016)	#14	Yi et al. (2017)
#7	Feng e Luo (2012)	#15	Urquiza-Fuentes e Paredes-Velasco (2017)
#8	Alnuaim, Caleb-Solly e Perry (2016)		

Tabela 5 – Identificação dos Artigos Selecionados

zagem que conduza ao desenvolvimento de capacidades de pensamento crítico”. Jeon (2015), por outro lado, defende que embora uma abordagem baseada em problemas pode não funcionar para iniciantes (i.e. alunos de graduação), “ele ainda são encorajados a se envolverem em todo processo de design, mas eles começam com um foco em uma parte do problema com orientação bem definida baseada na abordagem em camadas.”

Além das razões identificadas acima, a popularidade da aprendizagem ativa entre professores de IHC parece ficar clara ao assumir que essa abordagem aparentemente envolve os alunos de uma maneira que é mais próxima ao ambiente profissional. Muito da importância da IHC para a indústria de TI está relacionada com atividades práticas como prototipação e desenvolvimento de interfaces ou testes de usabilidade, por exemplo. As abordagens identificadas nos artigos selecionados buscam envolver ativamente os alunos em fazer algo ao invés de receberem informação passivamente. Para Hundhausen, Fairbrother e Petre (2012), oferecer aos alunos uma participação mais central “é semelhante à situação de um designer especialista que apresenta um design de interface de usuário a uma equipe de projeto em uma empresa do mundo real.”

RQ2 Quais são as ferramentas tecnológicas disponíveis, desenvolvidas especificamente para apoiar o ensino de IHC?

Embora a IHC esteja intimamente relacionada a computadores e outros tipos de dispositivos tecnológicos, o uso de ferramentas desenvolvidas especificamente para o ensino de IHC parece não estar muito disseminado. Considerando os artigos selecionados, apenas 5 mencionavam algum tipo de ferramenta durante o processo e, entre elas, apenas 2 usavam ferramentas especificamente desenhadas para o propósito.

Uma dessas ferramentas é um jogo sério, chamado UsabilityGame (BENITTI; SOMMARIVA, 2015), que oferece aos alunos a oportunidade de praticar o ciclo de vida de usabilidade, abordando análise de requisitos, prototipação e avaliação heurística. Ao jogá-lo, os alunos têm que desenvolver um protótipo para ser avaliado pelo professor e avaliar interfaces do mundo real escolhendo uma das heurísticas de Nielsen (NIELSEN, 1995) que melhoraria corretamente o problema de usabilidade detectado. O papel principal do professor, durante o jogo, é configurá-lo de acordo com os objetivos de aprendizagem dos alunos, conduzir a avaliação dos protótipos e monitorar o desempenho da turma.

A outra ferramenta chama-se WOZ Pro (Wizard of Oz Prototyper), um ambiente de prototipação de baixa fidelidade que foi desenvolvido para dar suporte à criação de protótipos e a testes wizard of oz (HUNDHAUSEN; FAIRBROTHER; PETRE, 2012). Embora tenha sido concebido para ser usado por alunos de graduação que estavam aprendendo design de interação pela primeira vez, ele também pode ser usado fora do contexto educacional (HUNDHAUSEN et al., 2007).

Algumas outras ferramentas com um ambiente de e-learning (SCELE – Student Centered e-Learning Environment (SANTOSO; SARI, 2015)), uma plataforma de mundo virtual (OpenSimulator (ZAHARIAS; BELK; SAMARAS, 2012)) ou até mesmo equipamento simples de arte (HUNDHAUSEN et al., 2007) foram citados como tendo sido usados durante o processo de ensino e eles todos podem ser tão valiosos quanto as ferramentas computacionais descritas acima. No entanto, professores de IHC parecem não ter muitas opções ao escolher alguma ferramenta que poderia atender aos seus propósitos educacionais.

RQ3 Como as metodologias ou tecnologias usadas para ensinar IHC têm sido avaliadas?

Os artigos selecionados também foram classificados seguindo a classificação de tipo de pesquisa proposta por Petersen, Vakkalanka e Kuzniarz (2015). De acordo com esse estudo, a maioria dos estudos primários (80%) foram classificados como artigos de experiência, apresentando as experiências dos autores em sala de aula. A importância desses artigos depende da relevância das lições aprendidas pelo autor a partir da sua experiência (WIERINGA et al., 2006). 2 outros estudos (13,3%) foram classificados como pesquisa de avaliação, com a proposta de novo conhecimento na área de educação de IHC. Finalmente, um estudo (6,7%) foi classificado como um artigo de opinião (ver Tabela 6).

Apesar da importância dos artigos de experiência, que contribuem com análises de experiências com metodologias na prática, revelando seus resultados quando aplicadas a contextos reais, o número de estudos avaliando novos desenvolvimentos no ensino de IHC é pequeno. O alto percentual de estudos de experiência com educação de IHC parece indicar que a maioria dos pesquisadores está preocupada em aplicar seus esforços para que entenda a área, produzindo dados que possam pavimentar a estrada para futuros desenvolvimentos.

Tipo de Pesquisa	Artigos
Artigo de experiência	#2, #4, #5, #6, #7, #8, #9, #10, #12, #13, #14, #15
Pesquisa de avaliação	#1, #3
Artigo de opinião	#11

Tabela 6 – Classificação de Tipo de Pesquisa

Ao avaliar o resultado das metodologias aplicadas, o questionário foi utilizado em 4 estudos (LAZEM, 2016; FENG; LUO, 2012; ALNUAIM; CALEB-SOLLY; PERRY, 2016; YI et al., 2017). 2 estudos aplicaram análise estatística em suas avaliações (Wilcoxon test (BENITTI; SOMMARIVA, 2015) and Pearson Product-Moment Correlation Coefficient (OR-BACH, 2015)). Entrevistas estruturadas foram usadas por Zaharias, Belk e Samaras (2012), enquanto que o estudo de caso foi usado por Santoso e Sari (2015) e por Hundhausen, Fairbrother e Petre (2012) e o estudo comparativo por Urquiza-Fuentes e Paredes-Velasco (2017).

Embora o ensino de IHC esteja presente em muitos programas de graduação relacionados à computação, parece que o tema não tem atraído suficiente atenção de pesquisadores, o que fica evidenciado pelo pequeno número de artigos selecionados. Esta conclusão fica patente verificando que por dois anos seguidos não houve publicações no tema. A falta de artigos selecionados dos anais de prestigiosas conferências mostra que o caminho ainda está aberto para a pesquisa na educação de IHC.

A maioria dos artigos selecionados apresentou pesquisas no ensino dos tópicos *Desenvolvimento* ou *Métodos de Avaliação*. Além disso, o ensino de alguns tópicos, como *Relevância da IHC* e *Dispositivos*, não foi sequer mencionado. A falta de pesquisa nesses tópicos não significa que eles são menos importantes, significa apenas que existe um campo ainda a ser explorado.

Metodologias ativas como um todo parecem ser a principal opção o ensino de IHC para alunos de graduação. O presente estudo não tentou categorizar a multitude de abordagens ou avaliar seus resultados quando aplicadas a esta área do conhecimento, por que a prevalência de uma abordagem sobre as outras não poderia ser atestada com o pequeno número de estudos primários encontrados. Essa categorização por eventualmente tornar-se objeto de um trabalho futuro.

Professores de IHC têm poucas opções quando têm que escolher uma ferramenta de suporte para suas aulas. Em um artigo de 2009, por exemplo, Wangenheim e Shull (2009) encontrou 16 jogos usados em educação de engenharia de software, a maioria deles baseados em computadores, mas também jogos de carta e tabuleiro. Embora o artigo conclua que mais pesquisa na área seja necessária, está claro que havia um número maior de jogos educacionais em engenharia de software 10 anos atrás do que há ferramentas educacionais para IHC hoje. Trabalhos futuros com essas ferramentas poderiam trazer grande benefício para professores e alunos de IHC.

4 Desenvolvimento

Como proposto no Capítulo 1, a seguir é detalhado o desenvolvimento de uma aplicação que possa servir como recurso didático no ensino do tema, apresentando aos alunos que a utilizarão a importância da observação de critérios de usabilidade no desenvolvimento de uma interface. Aqui são apresentados os requisitos do sistema, os casos de uso e tecnologias utilizadas, além do detalhamento de seu funcionamento, com algumas telas para facilitar a compreensão, informando os critérios de usabilidade que nortearam seu design.

Ao utilizar a aplicação, o aluno participará de dois estudos de caso. No primeiro estudo de caso, chamado aqui de Estudo de Caso 0 (EC0), será solicitado ao aluno realizar tarefas em um ambiente com usabilidade propositalmente reduzida. No segundo estudo de caso, chamado de Estudo de Caso 1 (EC1), as mesmas tarefas serão solicitadas ao aluno participante, mas, desta vez, as tarefas serão realizadas em um ambiente cujo design observa a utilização de critérios que aumentem a sua usabilidade. Após cada um dos estudos de caso, o aluno preencherá um formulário de satisfação SUS para, em seguida, receber um relatório com as respectivas medidas de eficiência, eficácia e satisfação resultantes da sua participação no estudo de caso. Cada um dos estudos de caso será independente um do outro, gerando seu próprio relatório.

4.1 Estudos de Caso

Quando executar os estudos de caso, o usuário deverá completar três tarefas distintas em ambientes com baixa e com alta usabilidade. Para que o usuário possa perceber as melhorias que uma interface com boa usabilidade proporciona para quem interage com ela, as tarefas solicitadas em EC0 e em EC1 serão as mesmas.

Ambos os estudos de caso ocorrerão sobre o mesmo sistema, um portal para divulgação e participação em eventos. O que diferencia cada estudo de caso é a interface utilizada por esse sistema em cada um deles. Assim, tanto em EC0, como em EC1, o usuário terá à sua disposição as mesmas funcionalidades, como, por exemplo, a relação dos eventos futuros ou a possibilidade de emissão de certificado de participação em eventos passados. No entanto, a forma como o usuário se relacionará com o sistema tende a ser bastante diferente nos dois estudos de caso, considerando que cada um terá uma interface diferente.

As tarefas a serem solicitadas são as seguintes:

Tarefa 1: Você pretende participar do próximo “Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais”. Descubra o valor de inscrição deste evento.

Tarefa 2: Você deve fazer a sua inscrição no “Simpósio Brasileiro de Qualidade de Software” utilizando os dados de pagamento que lhe serão disponibilizados ao iniciar a tarefa.

Tarefa 3: Você deve emitir o certificado de participação do Martin Fowler no último “Simpósio Brasileiro de Sistemas de Informação” utilizando os dados que lhe serão disponibilizados ao iniciar a tarefa.

4.2 Especificação dos Requisitos

Ao iniciar o desenvolvimento de uma aplicação de software deve-se saber o que ela precisa fazer para atender à necessidade proposta, pois “projetar e construir um programa de computador elegante que resolva o problema errado não serve às necessidades de ninguém” (PRESSMAN, 2010). Por isso, o desenvolvimento de qualquer aplicação inicia-se com a definição de seus requisitos, que são “as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento” (SOMMERVILLE, 2011).

A seguir são apresentados os requisitos da aplicação a ser desenvolvida. Como apresentado anteriormente, trata-se de um sistema que possibilitará ao usuário executar dois estudos de caso nos quais ele poderá perceber a diferença entre observar princípios de usabilidade ou não, e que serão desenvolvidos sobre o mesmo sistema, um portal para divulgação e participação em eventos. Dessa forma, serão desenvolvidos dois sistemas distintos: o sistema sobre o qual os estudos de caso serão executados, ou seja, o sistema de gerenciamento de eventos no qual o usuário deverá realizar as tarefas propostas, e que será chamado de “Portal de Eventos”, e o sistema que vai gerenciar os estudos de caso, controlando a interação do usuário com o sistema de eventos e apresentará as informações a respeito dessa interação, que será chamado de “UsabilidadeZero”.

Devido à natureza da aplicação, composta por dois sistemas, entendeu-se que seria necessário dividir a especificação dos seus requisitos. Dessa forma, são especificados os requisitos funcionais e as regras de negócio separadamente para a aplicação “UsabilidadeZero” e para o “Portal de Eventos”. Os requisitos não funcionais, ao contrário, serão especificados para todo o sistema, uma vez que ambos são executados simultaneamente sobre um mesmo ambiente computacional.

4.2.1 Aplicação UsabilidadeZero

4.2.1.1 Requisitos Funcionais

RF1.1: O sistema deve solicitar ao usuário identificar-se para que ele possa ter acesso aos seus estudos de caso e à visualização dos relatórios;

RF1.2: O sistema deve permitir ao usuário executar dois estudos de caso, nos quais o usuário deverá realizar três tarefas distintas em cada um:

- a) Estudo de Caso 0 (EC0): As tarefas a serem realizadas devem ser apresentadas em um ambiente com usabilidade reduzida;
- b) Estudo de Caso 1 (EC1): As tarefas a serem realizadas devem ser apresentadas em um ambiente cuja interface atenda a critérios de usabilidade, com ênfase nos critérios mais comuns na literatura identificados na Seção 2.2.2.

RF1.3: O sistema deve permitir ao usuário informar a conclusão ou desistência de uma tarefa e registrar se essa tarefa foi concluída com sucesso ou não;

RF1.4: O sistema deve registrar o tempo gasto pelo usuário e a quantidade de cliques que ele executa para a realização de cada tarefa;

RF1.5: O sistema deve permitir ao usuário responder ao questionário SUS referente à execução do EC0 ou do EC1, respectivamente;

RF1.6: O sistema deve permitir ao usuário visualizar um relatório com os resultados da execução do EC0 ou do EC1;

RF1.7: O sistema deve permitir ao usuário visualizar um relatório comparativo dos resultados da execução do EC0 e do EC1;

4.2.1.2 Regras de Negócio

RN1.1: O usuário deve poder identificar-se por meio de email;

RN1.2: O usuário deverá executar os dois estudos de caso, EC0 e EC1, em sequência e nessa ordem;

RN1.3: O resultado para cada estudo de caso deve contemplar as seguintes medidas de usabilidade:

- a) Medida de eficácia: percentual das tarefas concluídas com sucesso pelo usuário, além do percentual das tarefas que o usuário percebeu como concluídas;
- b) Medida de eficiência: tempo utilizado pelo usuário para a realização de todas as tarefas do estudo de caso e quantidade de cliques durante esse período;
- c) Medida de satisfação: resultado do questionário SUS respondido pelo usuário imediatamente após o término da última tarefa do estudo de caso.

RN1.4: O usuário deve poder confirmar o término da tarefa a qualquer momento após o seu início;

RN1.5: Para confirmar o término da tarefa, o usuário deve poder marcar sua conclusão ou sua desistência;

- a) Ao marcar a conclusão da tarefa, o usuário informa que conseguiu completá-la;
- b) Ao marcar a desistência da tarefa, o usuário informa que não conseguiu completá-la.

RN1.6: O sistema deve poder distinguir entre o término percebido (informado pelo usuário) e o término real de cada tarefa;

RN1.7: A contagem do tempo gasto e da quantidade de cliques necessários para completar cada tarefa deve iniciar-se quando o usuário aceita a tarefa, e parar quando o usuário confirma o seu término;

RN1.8: O tempo utilizado pelo usuário para a conclusão de cada estudo de caso deve ser a soma do tempo utilizado para a realização de cada uma de suas tarefas, assim como a quantidade de cliques executados deve ser a soma dos cliques executados em cada tarefa.

4.2.2 Portal de Eventos

4.2.2.1 Requisitos Funcionais

RF2.1: O sistema deve permitir ao usuário listar em sua página inicial todos os eventos cadastrados;

RF2.2: O sistema deve permitir ao usuário verificar as informações a respeito dos eventos;

RF2.3: O sistema deve permitir ao usuário inscrever-se nos eventos;

RF2.4: O sistema deve permitir ao usuário emitir o certificado de participação nos eventos.

4.2.2.2 Regras de Negócio

RN2.1: Todos os eventos cadastrados devem apresentar as seguintes informações: descrição, local, data e preço;

RN2.2: Para inscrever-se em um evento, o usuário deve informar seu nome, email e cpf, além dos dados para pagamento (número, validade e código de segurança do cartão de crédito);

RN2.3: O usuário somente poderá inscrever-se em eventos futuros;

RN2.4: O usuário somente poderá emitir certificado de participação em eventos passados.

4.2.3 Requisitos Não Funcionais

RNF1: O sistema deve ser uma aplicação web, utilizando o framework Play, implementado em código Java e/ou Scala;

RNF2: O sistema deve ser uma aplicação web compatível com o navegador Google Chrome 74;

RNF3: O sistema deve utilizar o banco de dados PostgreSQL 11.

4.3 Casos de Uso

Após o levantamento dos requisitos, foram definidos os casos de uso, que são processos de interação com o sistema (WAZLAWICK, 2011). Assim como os requisitos funcionais e as regras de negócio, também os casos de uso foram divididos entre os dois sistemas, considerados como sistemas de interação independentes. A seguir são apresentados os casos de uso de ambos os sistemas, “UsabilidadeZero” e “Portal de Eventos”.

4.3.1 UsabilidadeZero

A Figura 2 mostra o Diagrama de Casos de Uso para o sistema UsabilidadeZero:

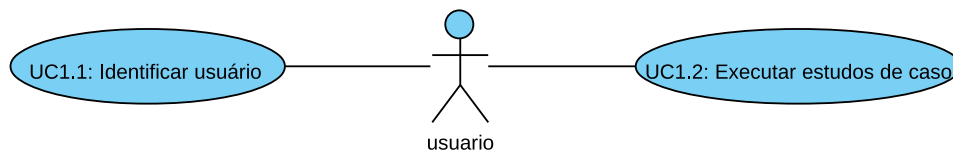


Figura 2 – Casos de Uso para a Aplicação UsabilidadeZero

UC1.1: Identificar usuário.

Objetivo: Identificar o usuário que faz o login para que ele possa ter acesso aos resultados de suas execuções de estudo de caso com os seus relatórios.

UC1.2: Executar estudos de caso.

Objetivo: Executar três tarefas em um ambiente cuja usabilidade foi propositalmente reduzida e em um ambiente que observa critérios de usabilidade; Aplicar questionário SUS para verificação da medida da satisfação do usuário após a sua execução do estudo de caso; Disponibilizar ao usuário um relatório com seu desempenho na tentativa de concluir as tarefas solicitadas após a execução de cada um dos estudos de caso e da resposta ao respectivo questionário SUS.

A seguir, o caso de uso UC1.2 é detalhado por meio de um diagrama de atividades para uma maior compreensão:

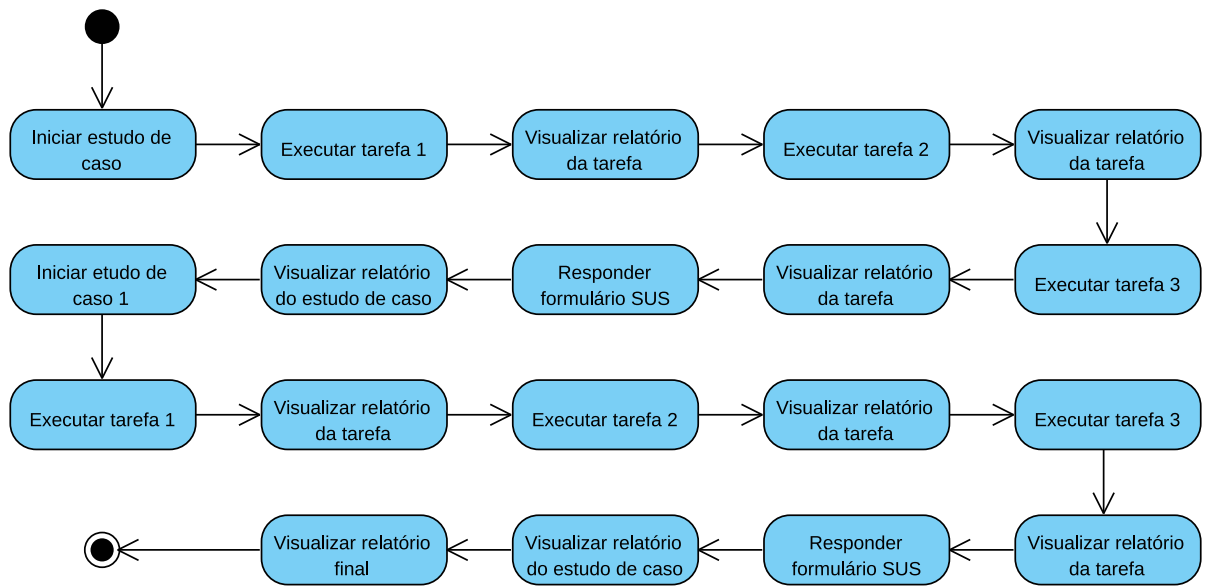


Figura 3 – Diagrama de atividades detalhando o caso de uso UC1.2

4.3.2 Portal de Eventos

A Figura 4 mostra o Diagrama de Casos de Uso para o sistema Portal de Eventos:

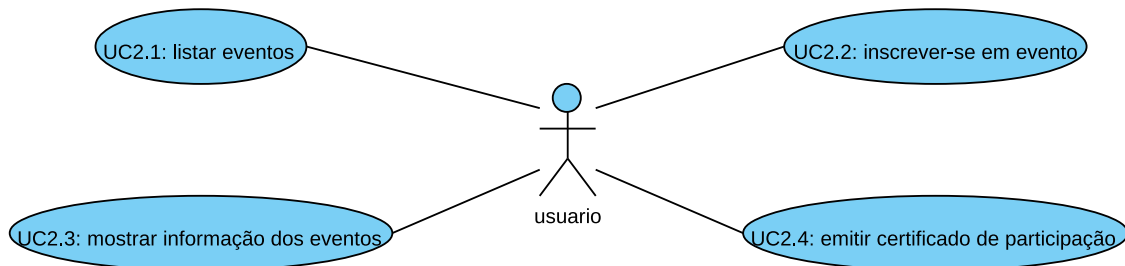


Figura 4 – Casos de Uso para a Aplicação Portal de Eventos

UC2.1: Usuário lista os eventos disponíveis no sistema.

Objetivo: Oferecer ao usuário a oportunidade de escolher o evento do qual deseja participar.

UC2.2: Usuário inscreve-se em evento.

Objetivo: Oferecer ao usuário a oportunidade de inscrever-se no evento do qual deseja participar.

UC2.3: Usuário verifica as informações sobre o evento.

Objetivo: Oferecer ao usuário as informações sobre o evento.

UC2.4: Usuário emite certificado de participação.

Objetivo: Permitir ao usuário emitir seu próprio certificado de participação no evento.

4.4 Aplicação

Toda a aplicação foi desenvolvida utilizando-se o framework Play, que utiliza, como padrão, a arquitetura MVC (*model-view-controller*). O Play permite que a camada do servidor seja programada tanto em Java quanto em Scala, uma vez que ambas as linguagens são interpretadas pela mesma máquina virtual (JVM). A camada cliente usa páginas HTML geradas dinamicamente pelo servidor utilizando o gerador de templates *template engine* Twirl a partir de *templates web*. Os templates são páginas web escritas em HTML, mas que também aceitam funções em Scala para processamento de parâmetros de entrada antes de serem compiladas e renderizadas como páginas HTML puras. Para acessar o banco de dados, o Play utiliza a biblioteca Slick, que utiliza conceitos de programação funcional da linguagem Scala para definir como se dá o acesso ao banco de dados.

O banco de dados utilizado foi o PostgreSQL 11, que foi escolhido para manter compatibilidade com o banco de dados em nuvem disponibilizado pelo provedor. O gerenciamento do banco de dados foi feito pelo Play, que utiliza uma tecnologia chamada Evolutions para gerar as tabelas. A cada alteração no esquema relacional o Play gera um arquivo SQL para destruir o antigo e gerar o novo.

4.4.1 Modelagem dos Dados

Para a representação dos elementos do domínio da aplicação (entidades) e seus relacionamentos e implementação do banco de dados foi feita uma modelagem dos dados conforme a Figura 5. Além das entidades tratadas neste documento (usuário, estudo, tarefa, etc.), há ainda as entidades `token_sistema`, para fazer controle do usuário e do estudo presentes na sessão do navegador, `token_cadastro`, para validação do cadastro do usuário, e `play_evolutions`, para gerenciamento das evolutions do Play.

4.4.2 Camada Cliente

A interface da aplicação foi desenvolvida utilizando as tecnologias tradicionais de desenvolvimento web, como HTML, CSS e JavaScript, além de Scala nos *templates web*. A utilização de uma linguagem como Scala em conjunto com o HTML facilita o processo de formação da árvore DOM, uma vez que ele possibilita desde a utilização de estruturas de programação (por exemplo, *if-else*, *for*, *switch-case*) a funções completas, bastando iniciar a expressão com o símbolo especial '@'.

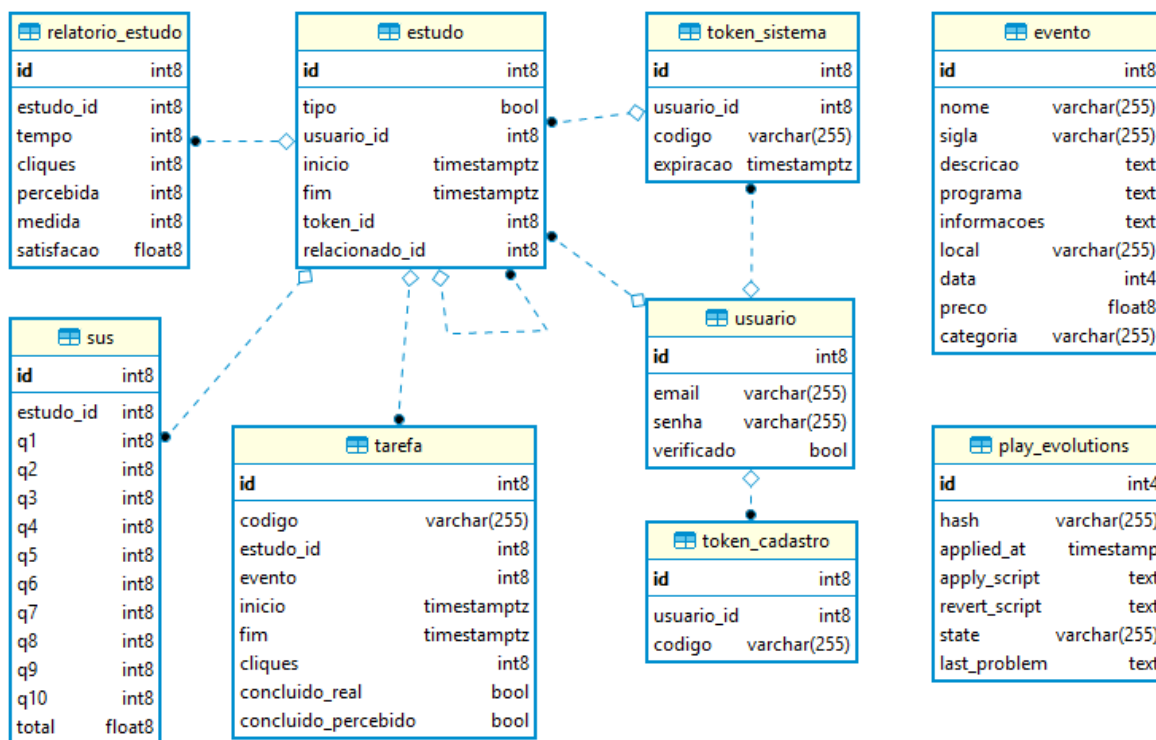


Figura 5 – Modelo Entidade-Relacionamento

A estilização da interface foi feita a partir do *framework front-end* Bootstrap 4, que serviu como padrão estético do sistema. O comportamento de alguns componentes desse framework, no entanto, foi individualmente modificados para maior adequação o sistema, como foi o caso da barra de caminho. O ‘Portal de Eventos’ teve praticamente toda a estilização retirada de seus elementos para dar a usuário uma sensação estética inferior.

A interatividade da interface foi programada em JavaScript, utilizando a biblioteca jQuery. Assim pode ser feito o controle das tarefas, como a contagem de cliques e a percepção de que um tarefa foi concluída. Outras bibliotecas que auxiliaram jQuery.appear, para detectar quando o valor do evento aparece na tela, jQuery.backDetect, para contar os cliques no botão ‘Voltar’ do navegador, e Isotope, para ordenação dos eventos na tela principal.

4.4.3 Camada Servidor

A aplicação foi hospedada no Heroku (<https://www.heroku.com>), que disponibiliza o serviço PaaS (*platform as a service*) em nuvem, e no qual foi possível a criação de uma conta gratuita que atendeu às necessidades da aplicação. A escolha pelo serviço se deu por ele oferecer suporte nativo ao Play, contando com o *sbt* já instalado, o que facilitou a implantação (*deploy*) do sistema, que foi feita diretamente do GitHub, que foi utilizado como repositório do código fonte da aplicação. O Heroku

também disponibiliza o serviço de banco de dados em nuvem como DBaaS (*database as a service*), tendo como banco de dados padrão o PostgreSQL.

4.5 Operacionalidade

A seguir é descrito o funcionamento da aplicação. A aplicação “UsabilidadeZero” e o “Portal de Eventos” são descritos separadamente.

4.5.1 UsabilidadeZero

Ao iniciar a aplicação, o usuário deve informar o seu email. Apesar de não ser necessária a realização de um cadastro, a informação do email do usuário foi importante no momento da avaliação da aplicação, pois ajudaria a relacionar os dados coletados durante os estudos de caso com os dados coletados no formulário de avaliação (ver Capítulo 5).

Logo após informar o seu email, recebe as primeiras instruções (Figura 6). Aqui, ele é informado a respeito do propósito da aplicação que ele está utilizando e o que se espera dele durante a utilização (completar as três tarefas solicitadas e preencher o formulário de satisfação). O usuário também é notificado que não é ele que está sendo avaliado, mas o sistema que ele utilizará.

Toda a utilização do aplicativo é feita de forma sequencial. Isso quer dizer que o usuário primeiramente vai executar o EC0, com suas três tarefas em ordem, e preencher o formulário SUS em sequência. Depois ele fará o mesmo com o EC1 para, no fim receber o relatório final com as medidas de usabilidade referentes à sua execução. Após cada passo, o usuário também recebe relatórios parciais (por tarefa ou por estudo de caso) para que ele possa acompanhar o andamento de sua execução.

Antes de iniciar cada estudo de caso, ele recebe uma breve instrução sobre como o estudo de caso está formatado (baixa ou alta usabilidade) e como deve proceder para confirmar o término da tarefa, seja porque ele entende que a concluiu, seja porque ele quer desistir da tarefa. A Figura 7 mostra as instruções para o EC0. Ao clicar no botão ‘Iniciar Estudo de Caso’, o usuário recebe a primeira tarefa e, logo em seguida, é direcionado ao “Portal de Eventos” para a sua execução.

Nas telas de instrução e dos relatórios dos estudos de caso, foi inserida uma barra de caminho, do tipo *multistep*, para que o usuário possa saber o quanto já foi feito e o quanto ainda falta para ele terminar os dois estudos de caso. Após o término de cada tarefa, o usuário recebe um relatório parcial informando o tempo decorrido do início ao fim da tarefa, a quantidade de cliques dados durante sua execução e se a tarefa foi, de fato, concluída ou não, o que pode ser visto na Figura 8.

Assim que a terceira tarefa de cada estudo de caso é concluída, é solicitado ao usuário que ele preencha o formulário SUS de satisfação. Seu preenchimento é rápido, bastando que o usuário

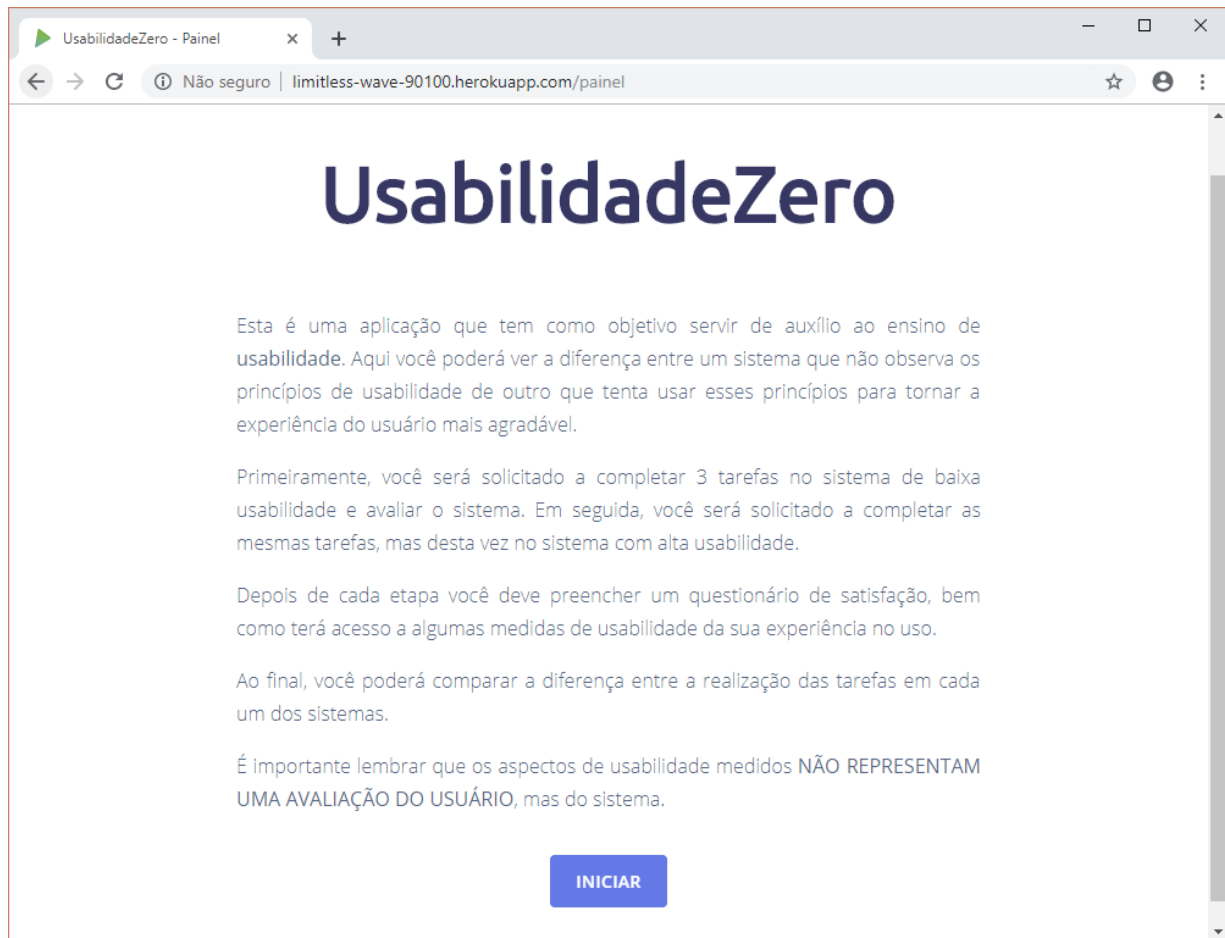


Figura 6 – Tela inicial

marque um valor na escala para cada uma das questões, variando de 1 (*Discordo totalmente*) a 5 (*Concordo plenamente*) (Figura 9).

Após o preenchimento do formulário SUS, o usuário recebe o relatório com as medidas de usabilidade na execução do estudo de caso recém concluído. O relatório apresenta as medidas por tarefa (tempo, cliques e eficácia percebida e medida) e também os valores para todo o estudo de caso, separados por medida de usabilidade. A medida de eficiência traz a quantidade de cliques e o tempo somados das três tarefas, a medida de eficácia apresenta a porcentagem de tarefas concluídas no estudo, e a medida de satisfação apresenta a pontuação obtida no formulário SUS respondido após o estudo de caso. A Figura 10 mostra o relatório após uma execução do EC0.

Concluídos os dois estudos de caso, com as três tarefas terminadas e preenchido o formulário SUS para ambos, é apresentado ao usuário um relatório final comparativo das medidas de usabilidade nos dois estudos de caso (Figura 11). Abaixo da tabela comparativa, o usuário recebe uma avaliação para que ele possa entender o que esses valores representam, dizendo em qual estudo de caso cada medida foi melhor.

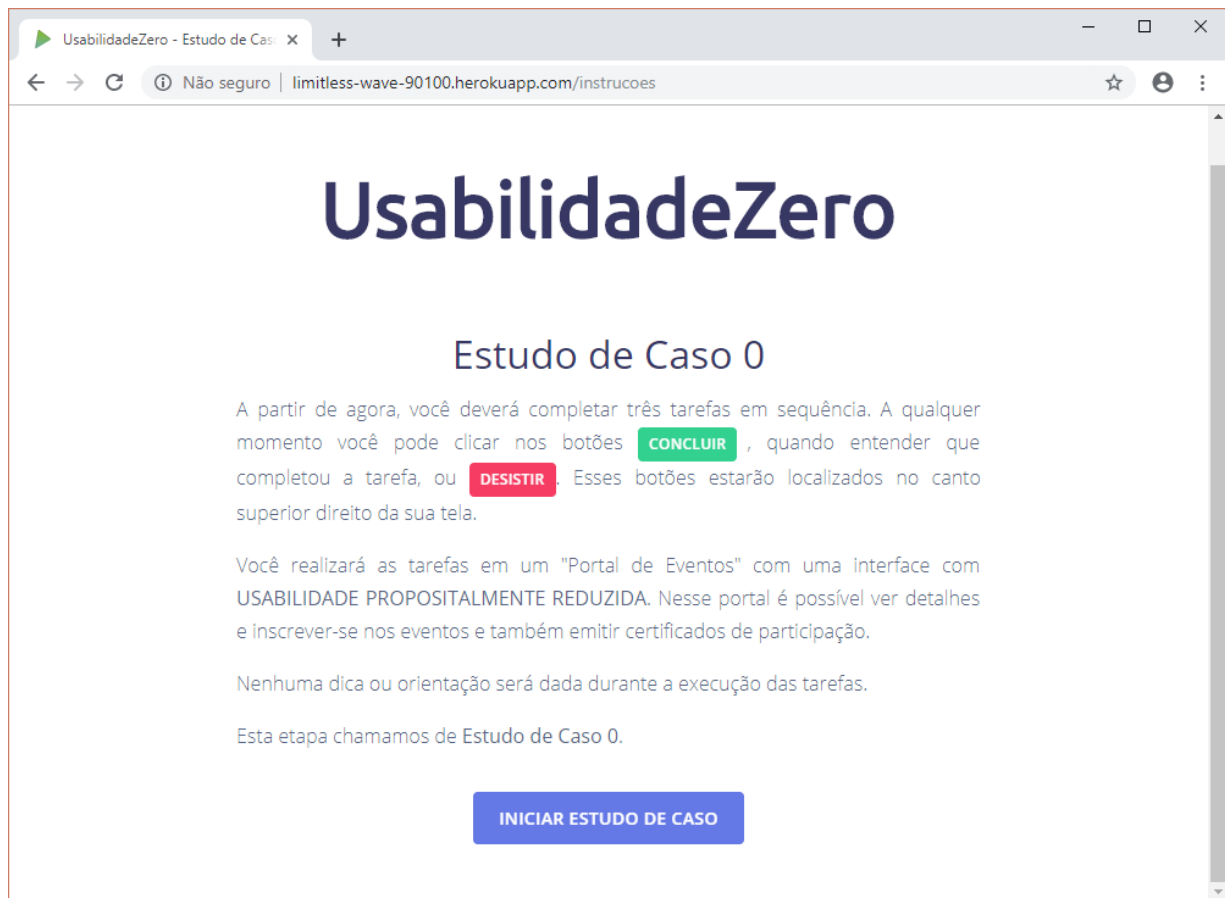


Figura 7 – Tela de instruções para o estudo de caso

4.5.2 Portal de Eventos

A interface do Portal de Eventos é composta por uma tela inicial, na qual os eventos são listados, e as telas de cada evento, com suas informações (data, local, valor, etc.) e formulários para inscrição, quando se tratar de evento futuro, e emissão de certificado, quando se tratar de evento passado. A diferença entre os dois estudos de caso está na aplicação dos critérios de usabilidade.

A seguir é apresentado o funcionamento do sistema, com suas telas principais nos dois estudos de caso. Em cada uma delas também são apresentados os critérios de usabilidade violados, no caso do EC0, ou atendidos, no caso do EC1.

Tela Inicial

A tela inicial do Portal de Eventos mostra todos os eventos cadastrados no sistema. Ao clicar em algum dos eventos, o usuário é levado à tela que mostra os detalhes daquele evento.

No EC0 (Figura 12), os eventos são listados sem ordenação definida, sejam eles eventos passados ou eventos futuros. Nessa tela, os eventos listados são identificados apenas por suas siglas.



Figura 8 – Tela de relatório após o término da tarefa



Figura 9 – Tela do formulário SUS (visão parcial)

UsabilidadeZero - Estudo de Cas x +

Não seguro | limitless-wave-90100.herokuapp.com/relatorio

Estudo de Caso 0

Resultado por tarefa

	Eficiência		Eficácia	
	Tempo	Cliques	Percebida	Real
Tarefa 1	32s	23	✓	✓
Tarefa 2	2min 4s	54	✗	✗
Tarefa 3	4min 46s	31	✓	✗

Resultado por medida de usabilidade

Eficiência	Cliques	108
	Tempo	7min 23s
Eficácia	Percebida	66%
	Verificada	33%
Satisfação		12.5 pts

PÁGINA INICIAL PRÓXIMO ESTUDO DE CASO

Figura 10 – Tela de relatório após o término do estudo de caso

As cores foram escolhidas de forma a dificultarem o contraste (fonte preta em fundo vermelho nos botões dos eventos e fonte amarela no fundo branco no título) e o título utiliza uma fonte cursiva que tende a dificultar a legibilidade.

Na tela inicial do EC1 mostrada na Figura 13, os eventos estão, inicialmente, ordenados por data. Outras formas de ordenamento são possíveis, como sigla, por nome ou por local do evento. Os eventos também foram separados em dois grupos: próximos eventos e eventos passados. Há ainda um campo para filtrar o evento que se deseja busca, bastando o usuário começar a digitar que os eventos que não possuem a expressão digitada serão ocultados. Nos campos dos eventos, suas principais informações são apresentadas para que o usuário possa rapidamente reconhecer o evento desejado.

Critérios de usabilidade violados no EC0:

- Redução da carga de trabalho: Os eventos são identificados apenas por suas siglas, levando o usuário a ter que clicar em cada um deles para saber mais informações, como nome e local, ou se trata-se de evento passado ou futuro;
- Adaptação ao usuário: Os eventos não estão ordenados em sequência lógica e não há caixa de busca.

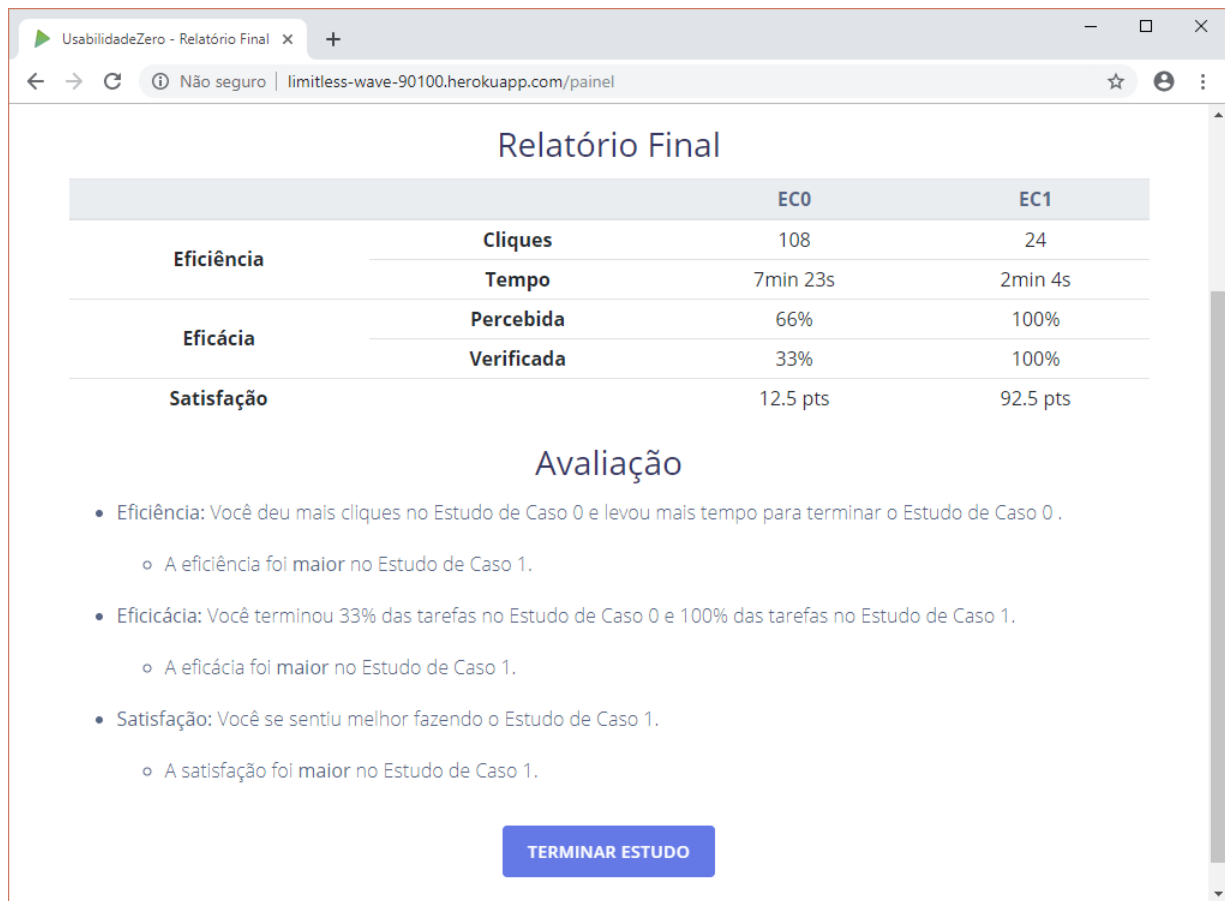


Figura 11 – Tela de relatório final

Critérios de usabilidade atendidos no EC1:

- Redução da carga de trabalho: As principais informações disponíveis e a caixa de busca facilitam ao usuário encontrar o evento desejado;
- Adaptação ao usuário: Os eventos podem ser reordenados de acordo com a necessidade do usuário.

Tela de Detalhes do Evento

A tela de detalhes do evento apresenta as informações gerais sobre aquele evento (do que se trata, tema do evento, periodicidade, etc.) retiradas da página da Sociedade Brasileira de Computação (SBC). A Figura 14 mostra a tela no EC0, que traz apenas essas informações mencionadas. Para mais informações, o usuário deve escolher em qual botão clicar no rodapé do texto. Os botões não estão dispostos em uma ordem lógica, o que dificulta ao usuário acostumar-se com sistema. Os textos dos botões também podem não ser muito claros para o usuário. O botão 'Pagamento', por exemplo, leva a uma tela que apresenta informações de como o pagamento pode ser feito, mas

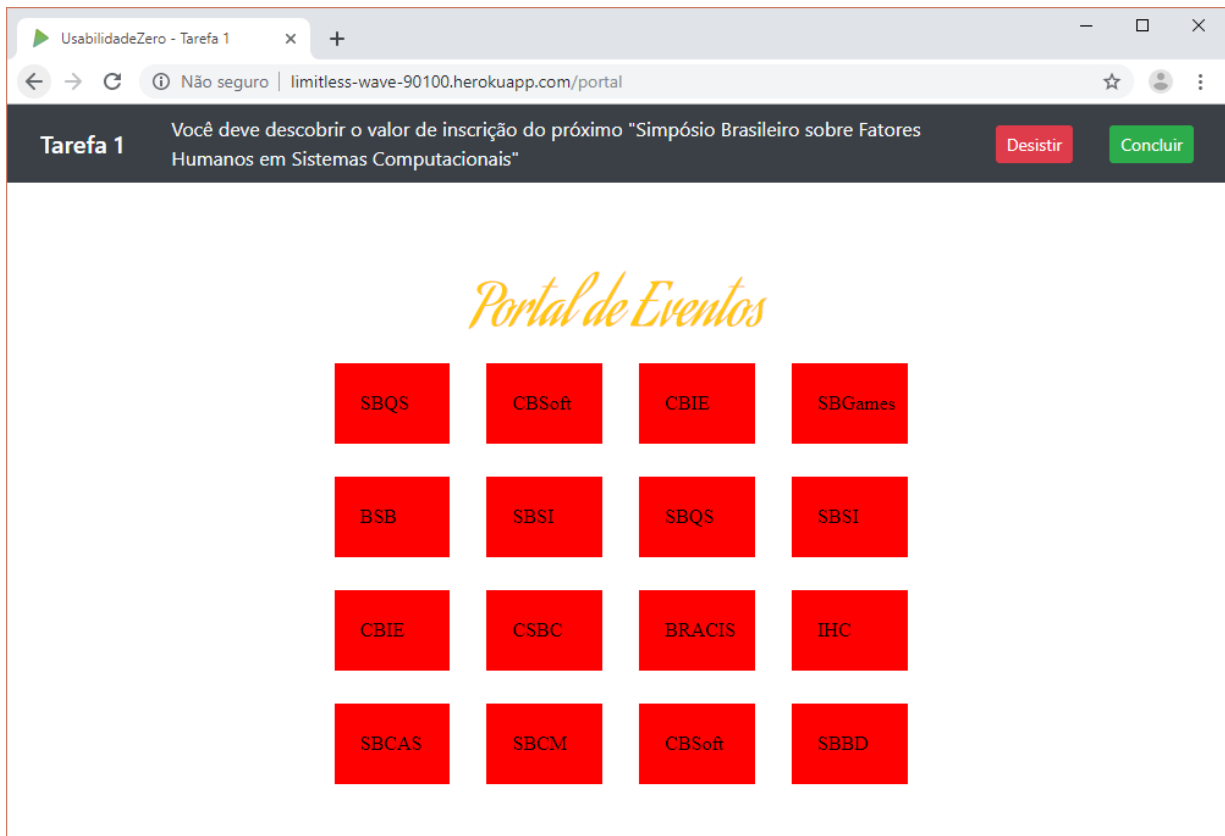


Figura 12 – Tela de inicial do Portal de Eventos no EC0

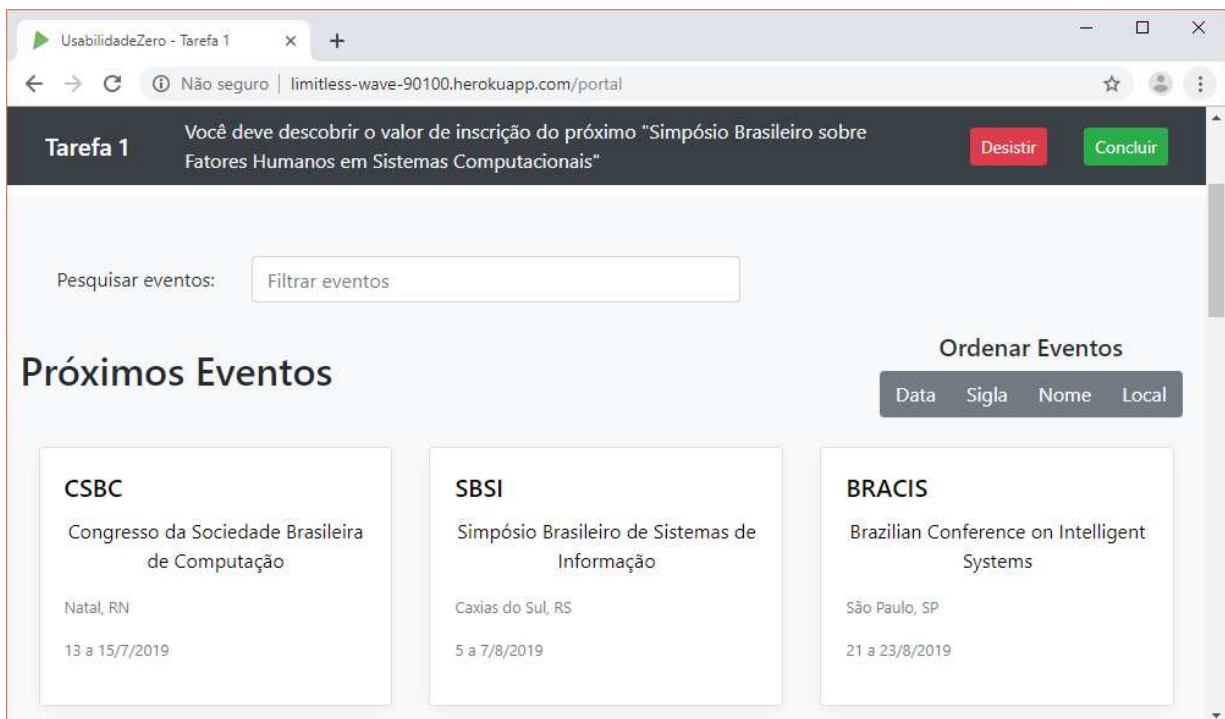


Figura 13 – Tela de inicial do Portal de Eventos no EC1

não informa o valor do evento. Os botões ‘Participe’ e ‘Certificado’ são mostrados em todos os eventos, passados ou futuros, o que não deixa claro se essas informações estão disponíveis ou não. Nenhuma das telas do evento apresenta um botão ‘Voltar’ ou ‘Cancelar’, levando ao usuário ter que fazer isso clicando no botão ‘Back’ do navegador.

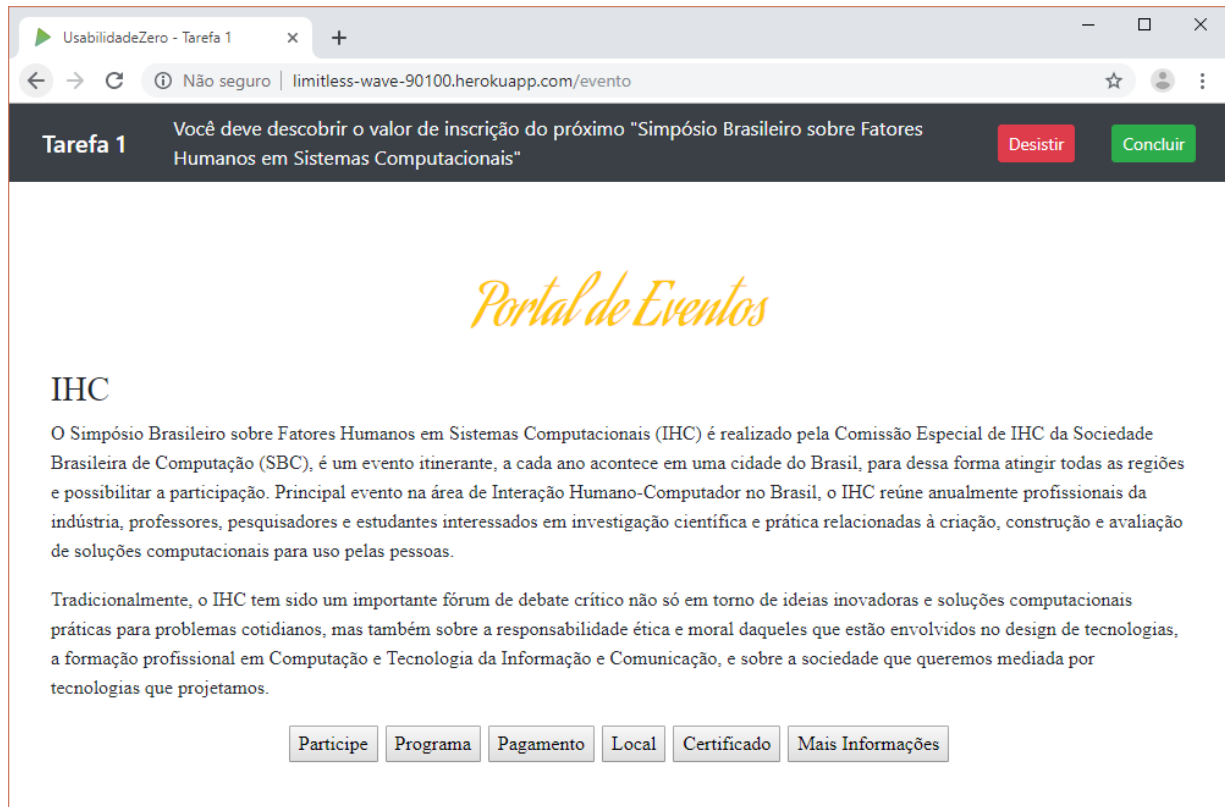


Figura 14 – Tela de detalhes de evento no EC0

Diferentemente do EC0, o EC1 apresenta todas as informações sobre o evento na mesma página, não sendo necessário escolher um botão para ser direcionado para outra página (Figura 15). Um menu lateral é disponibilizado para que o usuário possa ser levado diretamente à parte da tela que lhe interessa, mas, caso não seja esse seu interesse, ele pode simplesmente ver toda a página. Apenas o formulário para inscrição é apresentado em outra tela para maior clareza. De qualquer modo, o botão para inscrição no evento está disponível apenas nos eventos futuros, que ainda estão com inscrições abertas. Nos eventos passados, esse botão é substituído por um pequeno formulário de geração de certificado.

Crítérios de usabilidade violados no EC0:

- Redução da carga de trabalho: Necessita navegar entre as telas de informação dos eventos;

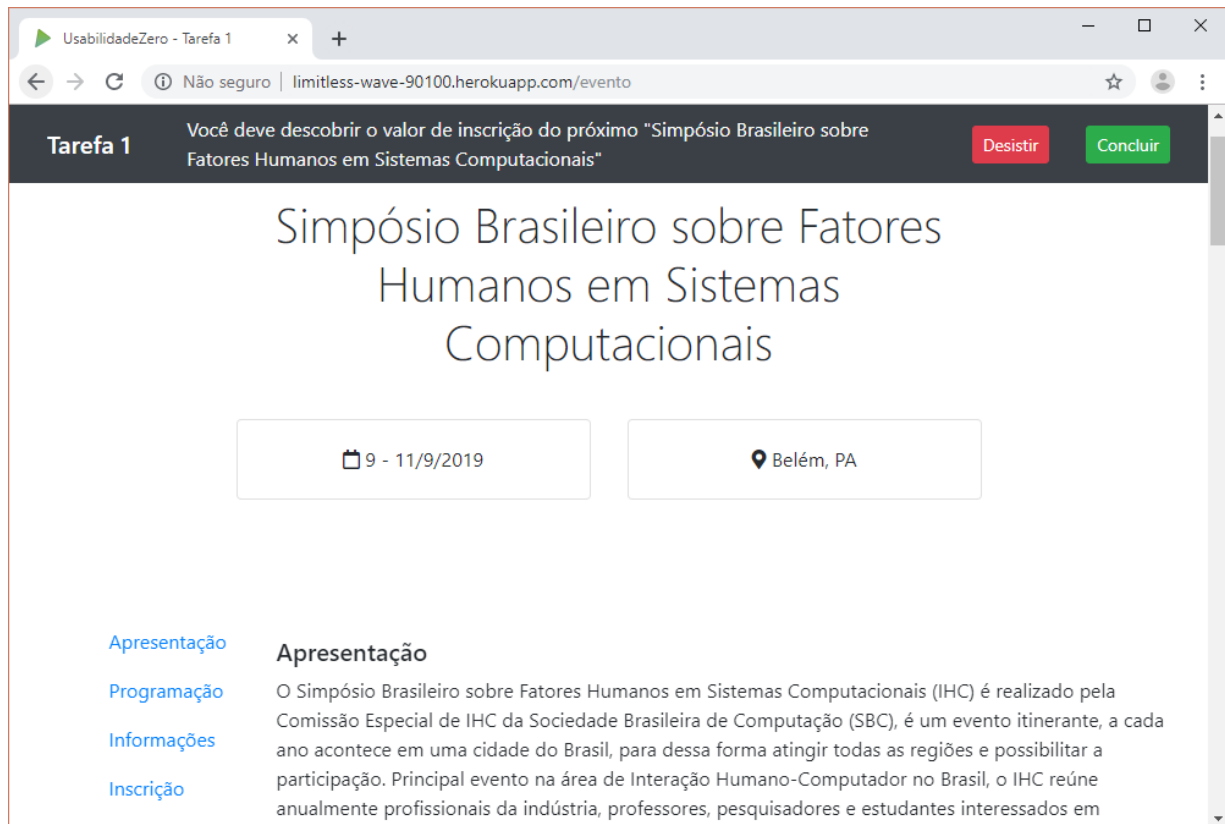


Figura 15 – Tela de detalhes de evento no EC1

- Consistência dos elementos da interface: As informações mostradas não condizem totalmente com o nome do botão;
- Consistência dos elementos da interface: Os botões não estão ordenados ou agrupados de maneira lógica;
- Tratamento de erros: O padrão de eventos passados e futuros é o mesmo, possibilitando confusão por parte do usuário, que pode tentar inscrever-se em evento passado ou tentar emitir certificado de evento futuro;
- Controle do usuário: Não há botões para voltar ou cancelar uma operação;

Critérios de usabilidade atendidos no EC1:

- Redução da carga de trabalho: Não é necessário que o usuário percorra várias telas para conseguir a informação desejada; Todas as informações do evento estão disponíveis na mesma tela;
- Consistência dos elementos da interface: Os *links* e as informações procuram utilizar termos adequados para evitar confusões por parte do usuário; Os botões ou *links* estão sempre na mesma ordem;

- Tratamento de erros: Eventos futuros permitem a inscrição e eventos passados permitem emitir certificado, não o contrário;
- Controle do usuário: Os botões ‘Voltar’ ou ‘Cancelar’ dão controle ao usuário;

Formulário de Inscrição em Evento

Por meio do formulário, o usuário se inscreve no evento desejado, fornecendo seus dados pessoais (nome, email, cpf) e de pagamento (dados do cartão). Esse formulário no EC0 (Figura 16) solicita dados que, neste caso, não estão relacionadas com a inscrição em si (fone, endereço e cidade) e que não são obrigatórias para o usuário. Os campos obrigatórios são marcados com a etiqueta (*label*) sublinhada, o que não é usual em campos desse tipo. Alguns campos (email, cpf, validade do cartão) exigem que os dados estejam devidamente formatados, mas essa informação e qual é o formato aceito não é informado ao usuário. Não há uma máscara no campo que leve o usuário a digitar corretamente. Caso haja algum erro, o formulário é recarregado com todos os campos em branco para que o usuário os digite novamente. O formulário não informa claramente qual foi o erro cometido. Por fim, a navegação por teclado (teclas ‘TAB’ e ‘Enter’) foi desabilitada, levando o usuário a utilizar apenas o mouse.

A Figura 17 mostra o formulário de inscrição no EC1. Esse formulário solicita apenas as informações que necessárias para a inscrição. Os campos obrigatórios são marcados com um asterisco, padrão amplamente utilizado para esse tipo de campo. Os campos que exigem algum formato pré-estabelecido utilizam máscara para evitar que o usuário digite incorretamente seus dados (cpf, número e validade do cartão) ou informam ao usuário que algo não está certo (email). Quando houver alguma falha na digitação ou campos não preenchidos, esses campos com problema são marcados para que o usuário perceba sua falha. O formulário conta ainda com os botões ‘Voltar’ e ‘Limpar Formulário’, além do ‘Enviar’, para dar mais controle ao usuário.

Critérios de usabilidade violados no EC0:

- Redução da carga de trabalho: Muitos dados para serem preenchidos, talvez desnecessariamente; Em caso de erro no preenchimento dos dados, a tela é reiniciada com os campos de texto vazios;
- Feedback ao usuário: A tela não mostra o valor que está sendo pago; Ao final da operação, a tela retorna para o início sem informar ao usuário se a operação foi concluída com sucesso ou não;
- Tratamento de erros: Não há máscara ou formatação pré-definida para os campos de texto;
- Adaptação ao usuário: A navegação por teclado (teclas ‘TAB’ e ‘Enter’) está desabilitada;

The screenshot shows a web browser window with the following content:

- Browser tab: UsabilidadeZero - Tarefa 2
- Address bar: Não seguro | limitless-wave-90100.herokuapp.com/evento/participe
- Header: Tarefa 2 | Você deve fazer a sua inscrição no "Simpósio Brasileiro de Qualidade de Software" utilizando os dados de pagamento disponíveis [aqui](#). Buttons: Desistir (red), Concluir (green).
- Form fields (all underlined):
 - Fone:
 - CPF:
 - Endereço:
 - Cidade:
 - # Cartão:
 - Titular do Cartão:
 - Validade:
 - CVV:
 - Forma de Pagto: Seleccione o número de parcelas (dropdown)
- Submit button: Participe
- Footer: Número do Cartão: 4609 8687 6694 4752 | Titular: "Escreva o seu nome" | Validade: 09/2030 | Código de Segurança: 902

Figura 16 – Formulário de inscrição em evento no EC0 (visão parcial)

- Controle do usuário: Não há botões para voltar ou cancelar a operação.
- Consistência dos elementos da interface: Os campos obrigatórios estão sublinhados, que é o padrão para *link* na web; Campos obrigatórios são, por padrão, marcados com um asterisco; Esta mudança tende a gerar confusão no usuário.

Critérios de usabilidade atendidos no EC1:

- Redução da carga de trabalho: Somente os dados relevantes devem ser preenchidos; Em caso de erro no preenchimento dos dados, somente os campos com problema devem ser corrigidos;
- Feedback ao usuário: A tela mostra o valor que está sendo pago; Ao final da operação, o usuário recebe uma mensagem informando o sucesso da operação;
- Tratamento de erros: Os campos de texto possuem máscaras para auxiliar o usuário no preenchimento;
- Adaptação ao usuário: Navegação por teclado habilitada;

UsabilidadeZero - Tarefa 2

Não seguro | limitless-wave-90100.herokuapp.com/evento/inscricao

Tarefa 2 Você deve fazer a sua inscrição no "Simpósio Brasileiro de Qualidade de Software" utilizando os dados de pagamento disponíveis [aqui](#). Desistir Concluir

Escreva o seu nome completo

Email*
nome@exemplo.com

CPF*
Informe o número do seu CPF

Pagamento

Número do Cartão* 4609 8687 6694 4752 Validade* 09/2030

Titular do Cartão* Código de Segurança*

Número de parcelas*
Selecione o número de parcelas

Voltar Limpar Formulário Enviar

Número do Cartão: 4609 8687 6694 4752 Titular: "Escreva o seu nome" Validade: 09/2030 Código de Segurança: 902

Figura 17 – Formulário de inscrição em evento no EC1 (visão parcial)

- Controle do usuário: O usuário pode desistir a qualquer momento, cancelando a operação.
- Consistência dos elementos da interface: Os campos obrigatórios estão marcados por asterisco, o que é padrão amplamente utilizado;

Emissão de Certificado de Participação em Evento

Para emitir um certificado de participação, o usuário deve preencher os campos correspondentes em um evento passado, como mostrado na Figura 18. No EC0, todos os eventos (passados e futuros) trazem o botão 'Certificado' e levam ao formulário adequado. Somente após o usuário preencher todo o formulário e enviá-lo, o sistema informa que não é possível emitir certificados de eventos que ainda não aconteceram, quando for esse o caso. Se os dados estiverem corretos e a participação for confirmada, o certificado é apresentado. Assim como no formulário de inscrição, os campos também exigem formatação própria dos dados (email e cpf) sem, no entanto, utilizar máscaras para

evitar erros ou apresentar informações adequados quando o erro ocorrer.

UsabilidadeZero - Tarefa 3

Não seguro | limitless-wave-90100.herokuapp.com/evento/certificadoForm

Tarefa 3 Você deve emitir certificado de participação no último "Simpósio Brasileiro de Sistemas de Informação" utilizando os dados disponíveis [aqui](#). Desistir Concluir

Portal de Eventos

SBSI - Certificado

Nome:

Email:

CPF:

Senha:

Nome: Martin Fowler **Email:** martin@martinfowler.com **CPF:** 683.563.770-64 **Senha:** 1234

Figura 18 – Formulário de emissão de certificado no EC0

Para que o usuário possa emitir o certificado de participação (Figura 19), ele deve preencher apenas dois campos: cpf e senha. Caso a participação no evento seja confirmada, o certificado é apresentado. Caso contrário, o usuário recebe uma mensagem informando o ocorrido. Os campos para solicitar o certificado estão visíveis apenas nos eventos passados. Nos eventos futuros, o usuário tem a opção de inscrição. O campo do cpf traz uma máscara de digitação para evitar erros.

Crítérios de usabilidade violados no EC0:

- Redução da carga de trabalho: Muitos dados para serem preenchidos, talvez desnecessariamente; Em caso de erro no preenchimento dos dados, a tela é reiniciada com os campos de texto vazios;
- Tratamento de erros: Não há máscara ou formatação pré-definida para os campos de texto;
- Controle do usuário: Não há botões para voltar ou cancelar a operação;
- Consistência dos elementos da interface: Os campos obrigatórios estão sublinhados, diferentemente da marcação por asterisco, amplamente utilizada.

UsabilidadeZero - Tarefa 3

Não seguro | limitless-wave-90100.herokuapp.com/evento#certificado

Tarefa 3 Você deve emitir certificado de participação no último "Simpósio Brasileiro de Sistemas de Informação" utilizando os dados disponíveis aqui. Desistir Concluir

Programação conhecimento)

Informações

- SI para governo (transparência, dados abertos, interoperabilidade, ecossistemas governamentais)
- Sistemas de apoio à decisão

Certificado

CPF*

Senha*

Voltar Gerar Certificado

Nome: Martin Fowler **Email:** martin@martinfowler.com **CPF:** 683.563.770-64 **Senha:** 1234

Figura 19 – Formulário de emissão de certificado no EC1

Critérios de usabilidade atendidos no EC1:

- Redução da carga de trabalho: Apenas dois campos para serem preenchidos;
- Tratamento de erros: Os campos de texto possuem máscaras para auxiliar o usuário no preenchimento;
- Controle do usuário: O usuário pode desistir a qualquer momento, cancelando a operação;
- Consistência dos elementos da interface: Os campos obrigatórios estão marcados por asterisco, o que é padrão amplamente utilizado.

5 Avaliação

Concluído o desenvolvimento da aplicação, precisava-se que ela fosse avaliada de forma sistemática e, para isso, decidiu-se pela utilização da abordagem GQM (*Goal-Question-Metric*), um método cujo resultado de sua aplicação traz a especificação de um sistema de medição visando um conjunto particular de questões e um conjunto de regras para a interpretação dos dados de medição (BASILI; CALDIERA; ROMBACH, 1994).

O modelo GQM é uma estrutura hierárquica definida de cima para baixo, começando com uma meta, que é refinada em perguntas e que, então, é refinada em métricas (Figura 20). A mesma métrica pode ser usada para responder diferentes perguntas (BASILI; CALDIERA; ROMBACH, 1994). O modelo resultante tem três níveis hierárquicos:

1. Nível conceitual (*goal*): uma meta para cada objeto a ser avaliado no contexto do projeto;
2. Nível operacional (*question*): um conjunto de perguntas para caracterizar como uma meta específica será alcançada;
3. Nível quantitativo (*metric*): um conjunto de dados associado a cada pergunta para respondê-la de forma quantitativa.

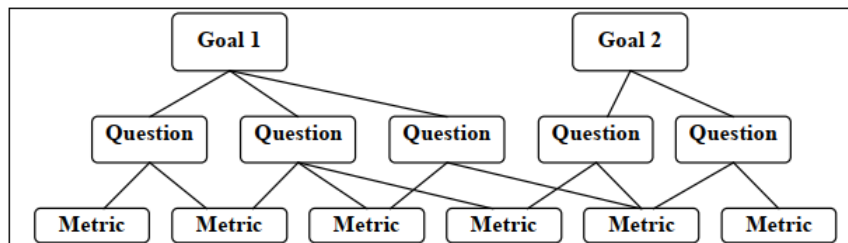


Figura 20 – Estrutura hierárquica do modelo GQM

Fonte: Basili, Caldiera e Rombach (1994)

A seguir é apresentado o planejamento da avaliação, definindo os objetivos a serem alcançados, com suas perguntas e métricas, e a elaboração dos instrumentos de medição, a execução da avaliação e os resultados obtidos que serão interpretados e discutidos ao final.

5.1 Planejamento

Ao avaliar a aplicação, pretendeu-se verificar se, de fato, ele trazia experiências distintas nos seus dois estudos de caso, ou seja, se o ECO proporcionava uma experiência de usabilidade reduzida

em relação a EC1, e se havia algum tipo de aprendizado sobre o tema por parte dos usuários após a sua utilização. Esses dois objetivos descritos de acordo com o template de definição de metas apresentado em Pressman (2010) da seguinte forma:

G1 Analisar o aplicativo UsabilidadeZero com o propósito de avaliar se ele permite experiências distintas em seus estudos de caso com respeito a medidas de usabilidade, do ponto de vista de alunos de cursos da área de Tecnologia da Informação no contexto da Engenharia de Usabilidade.

G2 Analisar o aplicativo UsabilidadeZero com o propósito de avaliar se ele permite aprender a respeito dos critérios e medidas de usabilidade do ponto de vista de alunos de cursos da área de Tecnologia da Informação no contexto da Engenharia de Usabilidade.

Para cada uma das metas definidas, foram associadas perguntas, cada qual com suas métricas. Para alcançar a meta G1, as perguntas Q1.1 a Q1.3 buscavam comparar ambos os estudos de caso (EC0 e EC1) com respeito às medidas de usabilidade (*eficiência, eficácia e satisfação*). Um última pergunta, Q1.4, visava verificar a satisfação do usuário em relação ao EC1.

Q1.1 O EC1 traz uma eficiência maior ao usuário quando comparado com o EC0?

M1.1.1 Tempo gasto por usuário em cada um dos estudos de caso;

M1.1.2 Número de cliques por usuário em cada um dos estudos de caso;

Indicador A eficiência é maior se a maioria dos usuários (mais de 50%) obtiver valores das métricas M1.1.1 e M1.1.2 no EC1 menores do que no EC0.

Q1.2 O EC1 traz uma eficácia maior ao usuário quando comparado com o EC0?

M1.2.1 Percentual de tarefas corretamente concluídas por usuário em cada um dos estudos de caso;

Indicador A eficácia é maior se a maioria dos usuários (mais de 50%) obtiver valores da métrica M1.2.1 no EC1 maiores do que no EC0.

Q1.3 O EC1 traz uma satisfação maior ao usuário quando comparado com o EC0?

M1.3.1 Pontuação no formulário SUS por usuário em cada um dos estudos de caso;

Indicador A satisfação é maior se a maioria dos usuários (mais de 50%) obtiver valores da métrica M1.3.1 no EC1 maiores do que no EC0.

Q1.4 O EC1 traz uma satisfação aceitável ao usuário?

M1.4.1 Média da pontuação dos usuários no formulário SUS no EC1;

Indicador A satisfação é aceitável se a média da pontuação de todos os usuários for acima de 70 pontos na escala SUS.

As perguntas Q2.1 a Q2.3, elaboradas para alcançar a meta G2, visavam verificar se houve algum aprendizado sobre usabilidade, especificamente quanto às medidas de usabilidade e quanto a critérios utilizados na construção dos estudos de caso. Por fim, a pergunta Q2.4 buscava a impressão dos alunos quanto a esse aprendizado.

Q2.1 Após o uso da aplicação, o usuário consegue reconhecer as medidas que comumente são utilizadas para avaliar a usabilidade de uma solução computacional?

M2.1.1 Respostas na questão “Quais aspectos são comumente verificados para avaliar a usabilidade de uma solução computacional?”.

Indicador Considera-se que o usuário consegue reconhecer as medidas se a maioria (mais de 50%) citar as *eficiência*, *eficácia* e *satisfação* em M2.1.1.

Q2.2 Após o uso da aplicação, o usuário consegue reconhecer como medir a usabilidade de uma solução computacional?

M2.2.1 Respostas na questão “Em um teste de usabilidade, como pode ser medida a EFICIÊNCIA de uma solução computacional?”;

M2.2.2 Respostas na questão “Em um teste de usabilidade, como pode ser medida a EFICÁCIA de uma solução computacional?”;

M2.2.3 Respostas na questão “Em um teste de usabilidade, como pode ser medida a SATISFAÇÃO de uma solução computacional?”;

Indicador Considera-se que o usuário consegue reconhecer como medir a usabilidade se a maioria (mais de 50%) obtiver respostas corretas em M2.2.1, M2.2.2 e M2.2.3.

Q2.3 O usuário consegue reconhecer critérios de usabilidade na sua experiência de uso da aplicação?

M2.3.1 Pontuação na questão “Eu consegui reconhecer diversos aspectos que impactam na facilidade de uso de uma aplicação” por usuário;

M2.3.2 Resposta na questão “Descreva detalhadamente 1 aspecto que impactou negativamente na sua experiência de uso” por usuário;

Indicador Considera-se que o usuário consegue reconhecer critérios de usabilidade se a maioria (mais de 50%) obtiver valores maiores do que 3 em M2.3.1 e a maioria citar algum critério adequado de usabilidade em M2.3.2.

Q2.4 O aplicativo UsabilidadeZero é uma solução “interessante” para aprender sobre usabilidade?

M2.4.1 Pontuação na questão “Eu percebi a importância da usabilidade para a qualidade de um produto de software” por usuário;

M2.4.2 Pontuação na questão “A experiência com o aplicativo vai contribuir para meu desempenho na vida profissional” por usuário;

M2.4.3 Pontuação na questão “Eu recomendaria este aplicativo para meus colegas” por usuário;

Indicador Considerado interessante se a maioria dos usuários (mais de 50%) obtiver valores maiores do que 3 em M2.4.1, M2.4.2 e M2.4.3.

As respostas às perguntas relativas ao objetivo G1 foram obtidas a partir do teste da aplicação feito pelos alunos, uma vez que os dados de utilização estavam guardados no seu banco de dados. Para obter as respostas às perguntas relativas ao objetivo G2 foi elaborado um formulário utilizando a ferramenta Formulários Google, disponível no Google Drive, e que pode ser visualizado no Apêndice A.

A fim de evitar situações que poderiam invalidar os dados (por exemplo, apenas um dos questionários preenchidos, questionários preenchidos fora de ordem, etc.), optou-se por manter todas as perguntas no mesmo formulário. Assim, cada aluno iniciaria a avaliação respondendo a algumas perguntas no formulário que, no momento certo, indicaria quando deveriam começar a utilizar a aplicação, com o seu respectivo *link*. Após o término do uso da aplicação, foi inserida uma notificação para que os alunos retomassem o formulário para responder às perguntas restantes.

No formulário, os alunos começariam respondendo à questão que solicitava sua concordância com a participação na avaliação, em seguida a um questionário de levantamento de perfil que buscava saber, por meio de perguntas diretas, se os alunos possuíam alguma experiência anterior com engenharia de usabilidade ou com testes de usabilidade. Em seguida, os alunos deveriam responder a perguntas para verificação de seu conhecimento prévio sobre o tema, mais especificamente sobre os três aspectos de usabilidade (*eficiência, eficácia e satisfação*) e suas formas de medição. Todas as perguntas eram obrigatórias.

As perguntas seguintes buscavam verificar o conhecimento dos alunos após o uso da aplicação. Essas perguntas retomavam as perguntas de conhecimento prévio para verificar o seu aprendizado. Em seguida, foram apresentadas perguntas sobre a aplicação. Excetuando a uma das perguntas (Pergunta 11, Apêndice A), que era aberta, todas as outras perguntas deveriam ser respondidas em uma escala Likert de 5 pontos, variando entre 1 (Discordo fortemente) e 5 (Concordo fortemente).

Todas as perguntas até aqui eram obrigatórias. Por fim, foi solicitado aos alunos que fizessem um comentário final (Pergunta 16) não obrigatório.

5.2 Execução

A avaliação da aplicação foi feita em uma amostra por conveniência de 32 alunos matriculados na disciplina Engenharia de Software do curso de graduação em Sistemas de Informação, no dia 28 de maio de 2019. Os alunos ainda não haviam trabalhado o tema *usabilidade* no semestre.

Durante o teste, todos os alunos utilizaram a aplicação simultaneamente e nenhuma falha foi detectada. Após o uso da aplicação, participando de ambos os estudos de caso, realizando suas tarefas e respondendo ao questionário de satisfação, os alunos retomaram as perguntas do formulário. Praticamente todos os alunos conseguiram responder aos testes sem maiores problemas. A exceção foi um aluno que, durante suas respostas no pós-teste, recarregou a página e o formulário reiniciou no pré-teste, perdendo suas respostas anteriores. Por esse motivo, seus dados foram descartados, reduzindo a amostra a 31 alunos.

Dos 31 alunos da amostra, 27 nunca haviam estudado o tema *usabilidade* (87,1%), enquanto 3 alunos já haviam feito algum curso ou disciplina (9,7%) e 1 já havia estudado o tema por interesse próprio (3,2%). Nenhum aluno trabalhava com o tema no campo profissional.

Quanto à experiência com testes de usabilidade, 27 alunos da amostra nunca haviam participado de qualquer teste de usabilidade (87,1%). Os outros 4 alunos já haviam participado de algum teste no papel de usuário do sistema avaliado (12,9%), sendo que apenas 1 deles havia informado na questão anterior que estudava o tema por interesse próprio. Os outros 3 responderam não ter conhecimento prévio.

5.3 Resultados

A partir dos dados coletados no uso da aplicação, foi possível responder às perguntas relativas ao objetivo G1. As respostas dadas no formulário puderam responder às perguntas relativas ao objetivo G2.

Q1.1) O EC1 traz uma eficiência maior ao usuário quando comparado com o EC0?

Para responder à pergunta Q1.1, foi medida a quantidade de cliques e o tempo gasto para completar cada estudo de caso, conforme estabelecido nas métricas M1.1.1 e M1.1.2.

A Figura 21 mostra a quantidade de cliques por usuário em cada estudo de caso. Na média, os usuários deram mais cliques no EC0 (102,8 cliques) do que no EC1 (41,64 cliques), o que representou uma redução média de 59,5% na quantidade de cliques. Entre os 31 participantes, 3 deles

deram mais cliques no EC1 e um participante deu a mesma quantidade de cliques. Assim, 87% dos usuários tiveram redução na quantidade de cliques no EC1.

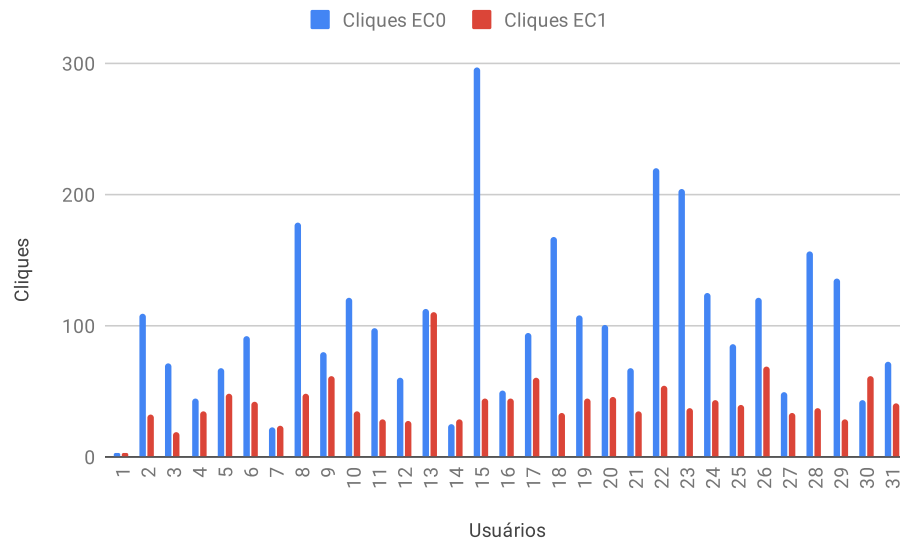


Figura 21 – Quantidade de cliques por usuário em cada um dos estudos de caso

O tempo gasto em cada um dos estudos de caso por usuário pode ser verificado na Figura 22. A média de tempo gasto no EC1 (204,5 segundos) foi 62,5% menor do que no EC0 (545,8 segundos). Apenas 2 participantes levaram mais tempo para terminar o EC1, ou seja, 93,5% terminaram o EC1 em menos tempo do que o EC0.

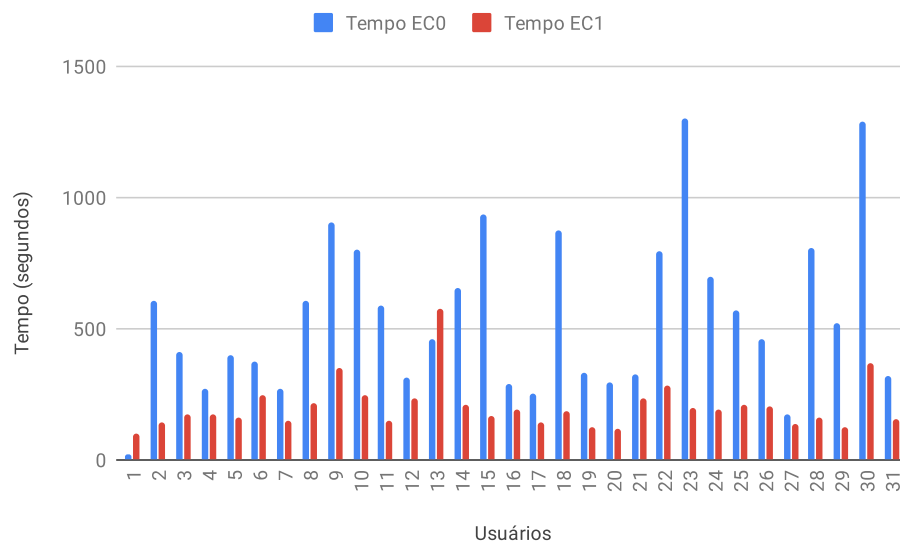


Figura 22 – Tempo por usuário em cada um dos estudos de caso

Com os valores correspondentes à quantidade de cliques e ao tempo gasto para terminar os estudos de caso por cada usuário, é possível agora responder à Q1.1, com base no indicador de suas métricas. Como tanto em M1.1.1 quanto em M1.1.2 os valores são menores para EC1 do que para EC0, a Q1.1 é respondida positivamente, afirmando que “O EC1 traz uma eficiência maior ao usuário quando comparado com o EC0”.

Q1.2) O EC1 traz uma eficácia maior ao usuário quando comparado com o EC0?

A métrica M1.2.1 definiu como medida o percentual de tarefas completadas corretamente por usuário em cada um dos estudos de caso para que seja respondida a pergunta Q1.2.

Nenhum usuário participante concluiu mais tarefas no EC0 do que no EC1, como pode ser verificado na Figura 23. Ao comparar os dois estudos de caso, vê-se que 20 alunos (64%) completaram mais tarefas no EC1 do que no EC0, embora 2 deles completaram apenas 1 tarefa no EC1 depois de não completarem nenhuma no EC0. 11 usuários concluíram a mesma quantidade de tarefas no dois estudos de caso.

Ao analisar separadamente cada um dos estudos, verifica-se que no EC0 apenas 3 usuários (9,7%) conseguiram completar as 3 tarefas e 8 (25,8%) não completaram nenhuma, enquanto que no EC1 14 usuários (42,2%) terminaram todas as tarefas e apenas 1 (3,2%) usuário não completou nenhuma.

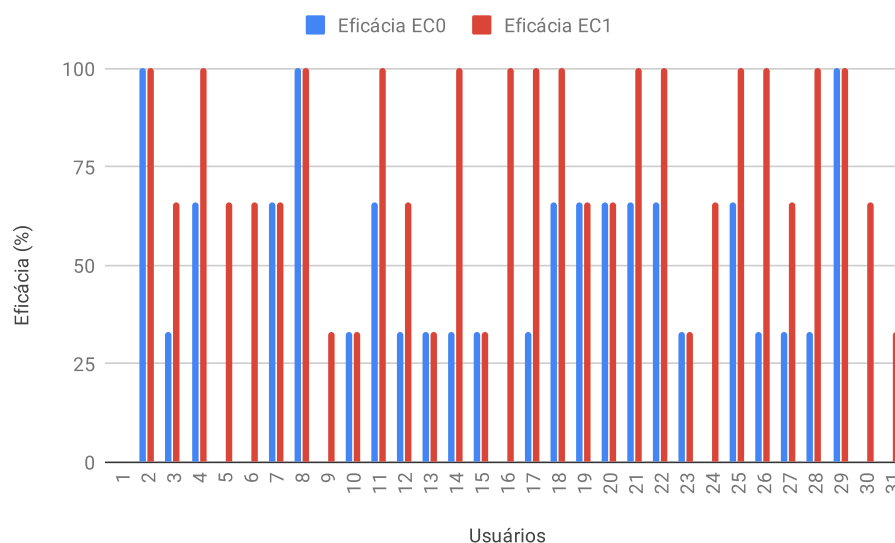


Figura 23 – Eficácia por usuário em cada um dos estudos de caso

A partir dos percentuais de tarefas concluídas nos dois estudos de caso, obtém-se a métrica M1.2.1 e, seguindo o indicador de que mais de 50% dos usuários com um percentual de tarefas completadas no EC1 do que no EC0, é possível afirmar que “O EC1 traz uma eficácia maior ao usuário quando comparado com o EC0”.

Q1.3) O EC1 traz uma satisfação maior ao usuário quando comparado com o EC0?

A resposta à pergunta Q1.3, conforme a métrica M1.3.1, foi dada pela medição da satisfação dos usuários em relação a cada um dos estudos de caso feita por meio do preenchimento do formulário SUS e a verificação de sua pontuação. Excluindo dois usuários que obtiveram pontuação igual para os dois estudos de caso (50 pontos), todos os usuários apresentaram satisfação maior no EC1 do que no EC0, o que pode ser verificado na Figura 24.

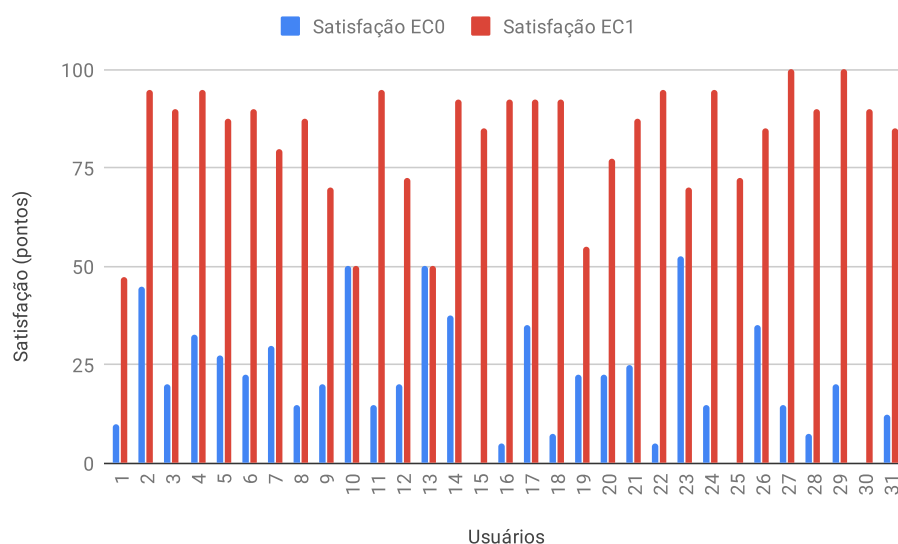


Figura 24 – Satisfação por usuário em cada um dos estudos de caso

Com os valores de satisfação medidos nos estudos de caso, correspondendo à métrica M1.3.1, é possível apresentar uma resposta positiva à Q1.3, afirmando que “O EC1 traz uma satisfação maior ao usuário quando comparado com o EC0”.

Q1.4) O EC1 traz uma satisfação aceitável ao usuário?

Seguindo a métrica M1.4.1, foi calculada a média da pontuação dos usuários no formulário SUS após o EC1 para responder à pergunta Q1.4. A satisfação no EC0 foi, em média, de 21,8 pontos e no EC1 a média foi de 82,8 pontos, indicando que nesse estudo de caso os usuários obtiveram um nível de satisfação aceitável de acordo com a escala SUS. No EC1, 4 usuários obtiveram pontuação abaixo de 70 e outros 4 obtiveram pontuação entre 70 e 75 (12,9% cada). Dos outros usuários restantes (74,2%), 2 (6,4%) obtiveram 100 na escala SUS (Figura 24), o que corresponde ao maior valor de satisfação imaginável (ver Figura ??).

Com o valor da métrica M1.4.1, a pergunta Q1.4 pode ser respondida positivamente, afirmando que “o EC1 traz uma satisfação aceitável ao usuário”.

Q2.1) Após o uso da aplicação, o usuário consegue reconhecer as medidas que comumente são utilizadas para avaliar a usabilidade de uma solução computacional?

A métrica M2.1.1 definiu como a pergunta Q2.1 deveria ser respondida. Conforme pode ser verificado no gráfico da Figura 25, antes do uso da aplicação, 13 (41,9%) alunos reconheceram *eficiência* como uma medida de usabilidade, 15 alunos (48,4%) reconheceram *eficácia* e 16 alunos (51,6%) reconheceram *satisfação*. Após o uso, esses valores aumentaram para 24 (77,4%), 27 (87,1%) e 25 (80,6%) alunos, respectivamente.

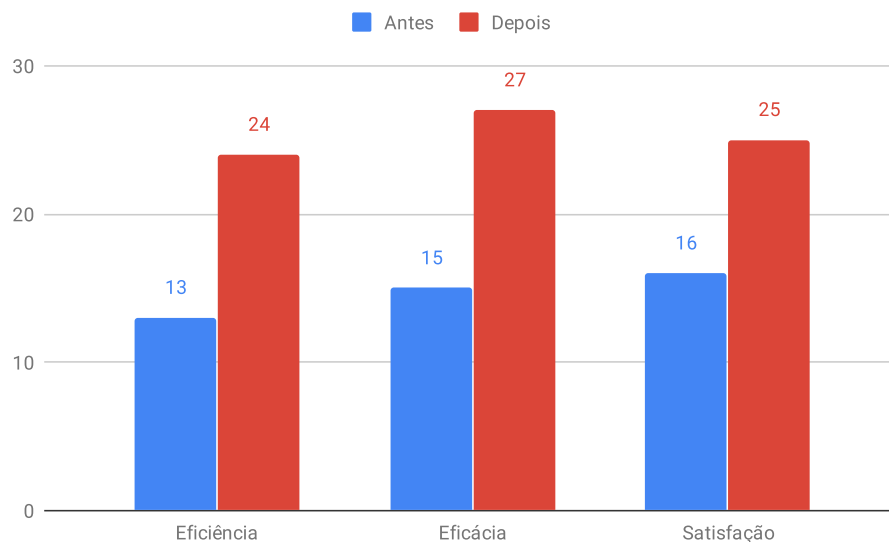


Figura 25 – Reconhecimento das medidas de usabilidade antes e depois do uso do aplicativo

Os valores da métrica M2.1.1 mostram o aumento do reconhecimento das medidas após o uso da aplicação e permitem uma resposta positiva à pergunta Q2.1, na qual pode ser afirmado que “após o uso da aplicação, o usuário consegue reconhecer as medidas que comumente são utilizadas para avaliar a usabilidade de uma solução computacional”.

Q2.2) Após o uso da aplicação, o usuário consegue reconhecer como medir a usabilidade de uma solução computacional?

A resposta à pergunta Q2.2 é elaborada a partir das métricas M2.2.1, M2.2.2 e M2.2.3, que mensuram o quanto os alunos conseguem reconhecer como medir os três aspectos (*eficiência*, *eficácia* e *satisfação*) da usabilidade de um sistema.

Ao serem perguntados sobre como medir cada um desses aspectos de usabilidade antes de utilizarem o aplicativo, 10 (32,3%) usuário responderam corretamente a respeito da *eficiência*, 10 (32,3%) corretamente a respeito da *eficácia* e 16 (51,6%) corretamente a respeito da *satisfação*.

Respondendo à mesma pergunta após o uso do aplicativo, 19 (61,3%), 16 (51,6%) e 19 (61,3%) alunos, respectivamente, apresentaram resposta correta, de acordo com o gráfico da Figura 26.

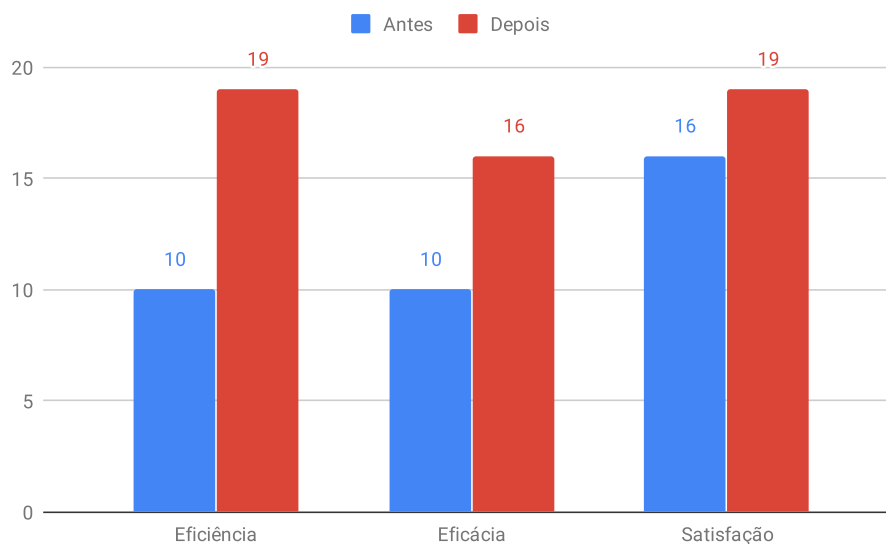


Figura 26 – Reconhecimento de como medir a usabilidade antes e depois do uso do aplicativo

Como mais de 50% dos alunos obtiveram respostas corretas em M2.2.1, M2.2.2 e M2.2.3, a resposta positiva à Q2.2 é que “após o uso da aplicação, o usuário consegue reconhecer como medir a usabilidade de uma solução computacional”.

Q2.3) O usuário consegue reconhecer critérios de usabilidade na sua experiência de uso da aplicação?

As métricas M2.3.1 e M2.3.3 trazem os dados que ajudam a responder à pergunta Q2.3. Ao responder se conseguiram reconhecer diversos aspectos que impactam na facilidade de uso de uma aplicação, 26 alunos (83,9%) apresentaram valores 4 ou 5, conforme gráfico da Figura 27. Quando solicitados a descrever um aspecto que impactou negativamente na sua experiência de uso, 25 alunos (80,6%) citaram algum critério de usabilidade, conforme mapeamento apresentado no Apêndice B.

Assim, utilizando as métricas M2.3.1 e M2.3.3, a resposta afirmativa à pergunta Q2.3 é que “O usuário consegue reconhecer critérios de usabilidade na sua experiência de uso da aplicação”.

Q2.4) O aplicativo UsabilidadeZero é uma solução “interessante” para aprender sobre usabilidade?

Para avaliar o interesse dos usuário pela aplicação e responder à pergunta Q2.4, foram estabelecidas três métricas, verificando a pontuação na escala Likert de 5 pontos nas respectivamente nas três

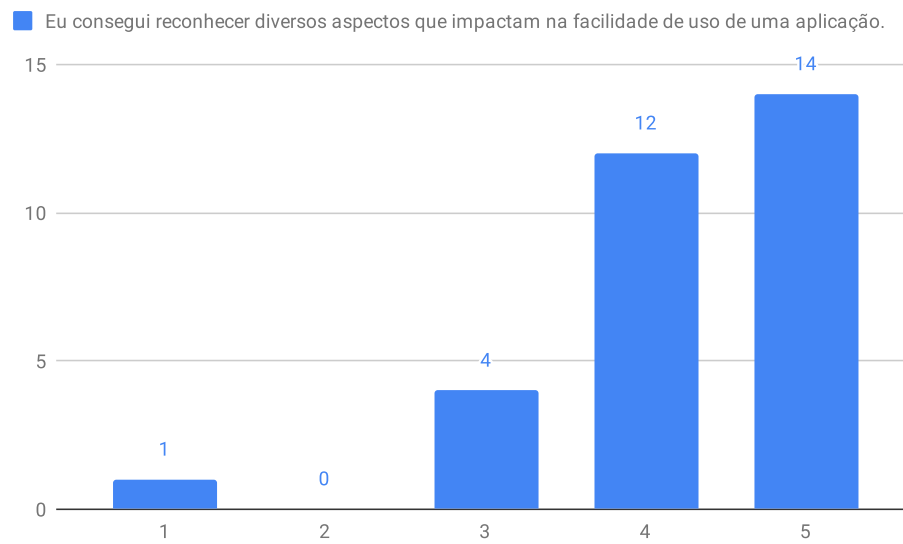


Figura 27 – Reconhecimento de critérios de usabilidade

questões a seguir:

- M2.4.1 Pontuação na questão “Eu percebi a importância da usabilidade para a qualidade de um produto de software” por usuário;
- M2.4.2 Pontuação na questão “A experiência com o aplicativo vai contribuir para meu desempenho na vida profissional” por usuário;
- M2.4.3 Pontuação na questão “Eu recomendaria este aplicativo para meus colegas” por usuário;

Assim, a aplicação seria considerada interessante caso a maioria dos usuários (mais de 50%) obtivesse valores maiores do que 3 em M2.4.1, M2.4.2 e M2.4.3.

O gráfico da Figura 28 mostra as respostas às métricas M2.4.1, M2.4.2 e M2.4.3. Por esse gráfico, percebe-se que ampla maioria respondeu acima de 3 (93,5%), com 2 respostas 4 (6,5%) e 27 respostas 5 (87,1%). Ainda houve 2 respostas 3 (6,5%). Nenhum usuário respondeu 1 ou 2.

À pergunta da métrica M2.4.2, 25 usuários responderam acima de 3 (80,6%), dos quais 12 responderam 4 (38,7%) e 13 responderam 5 (41,9%). Dos 6 usuários restantes, houve 1 resposta 1 (3,2%), 2 respostas 2 (6,5%) e 3 respostas 3 (9,7%).

Por fim, 25 usuários (80,6%) responderam acima de 3 à pergunta da métrica M2.4.3. Desses, 5 responderam 4 (16,1%) e 20 responderam 5 (64,5%). Houve ainda 1 resposta 1 e 1 resposta 2 (3,2% cada) e 4 respostas 3 (12,9%).

Considerando que nas três métricas mais da metade dos alunos responderam acima de 3 (93,5%, 80,6% e 80,6%, respectivamente), é possível responder positivamente à pergunta Q2.4, afirmando que “O aplicativo UsabilidadeZero é uma solução ‘interessante’ para aprender sobre usabilidade”.

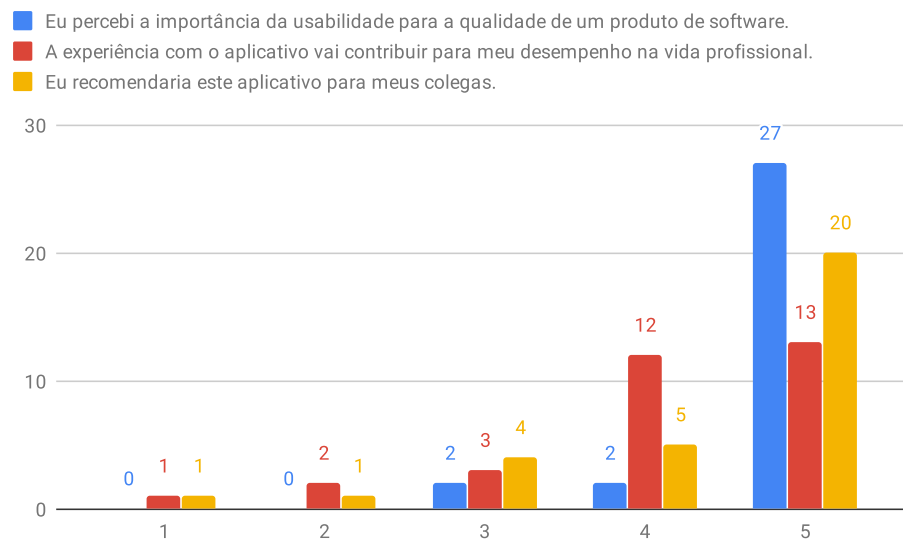


Figura 28 – Métricas M2.4.1, M2.4.2 e M2.4.3

5.4 Discussão

Após a avaliação, com todas as questões elaboradas no planejamento respondidas, pode-se verificar se as metas foram alcançadas.

A meta G1 buscava avaliar se a aplicação trazia experiências distintas nos seus dois estudos de caso. Para alcançar essa meta, quatro perguntas foram formuladas (Q1.1 a Q1.4). Todas essas perguntas receberam respostas afirmativas.

A pergunta Q1.1 investigava a diferença de eficiência nos dois estudos de caso trazidos pelo aplicativo. Para verificação da eficiência, foram medidos tanto o tempo para conclusão das tarefas, quanto a quantidade de cliques, que deveriam apresentar valores menores em EC1 quando comparado com EC0. A redução nas duas medidas foi acentuada. A quantidade média de cliques reduziu de 102,8 no EC0 para 41,64 no EC1 (redução de 56,5%) e o tempo médio foi de 545,8 segundos no EC0 para 204,5 segundos no EC1 (redução de 62,5%). Nos extremos, a menor quantidade de cliques em EC1 foi 3 e a maior 110 no EC1 e o tempo variou entre 101 segundos e 574 segundos no mesmo estudo de caso.

Na avaliação da diferença de eficácia entre os dois estudos de caso com a pergunta Q1.2, a primeira coisa que se percebe é que não houve aluno que completou mais tarefas no EC0 do que no EC1, sendo que 20 alunos (64% da amostra) completaram mais tarefas no EC0 do que no EC1. No entanto, considerando que esse é um ambiente onde os critérios de usabilidade foram aplicados, acredita-se que a quantidade de alunos que conseguem completar as três tarefas deveria ser próximo de 100%, o que não ocorreu, ficando essa taxa em 42,2%. Ao analisar separadamente cada tarefa nesse estudo de caso, a tarefa 1 foi concluída por 27 alunos (87,1%), a tarefa 2 por 25 alunos

(80,6%) e a tarefa 3 por 17 alunos (54,8%).

Esses números mostram que o ambiente do EC1 ainda precisa de melhorias, principalmente em relação à tarefa 3, onde os alunos deveriam emitir um certificado de participação em um evento passado. A partir da observação da interação dos usuários com o sistema, percebeu-se que muitos deles não encontraram os eventos passados, que estavam após os eventos futuros, no final da página. Uma indicação desses eventos no início da página (um botão, um link ou uma aba, por exemplo) poderia minimizar esse problema e aumentar a eficácia dessa tarefa.

A diferença entre a satisfação proporcionada pelos dois estudos de caso, analisada pelas perguntas Q1.3 e Q1.4, também foi bastante acentuada. Enquanto que no EC0 a média da pontuação dos alunos na escala SUS ficou em 21,8 pontos, no EC1 foi de 82,8 pontos, aproximando-se de uma satisfação excelente (BANGOR; KORTUM; MILLER, 2008). Dos dois alunos que obtiveram 50 pontos nos dois estudos de caso, um deles marcou sempre os mesmos valores.

Baseado nas respostas apresentadas acima, pode-se concluir que o estudo de caso 1 apresenta ao usuário uma usabilidade maior quanto comparada com o estudo de caso 0, proporcionando-lhe experiências distintas que podem lhe auxiliar a perceber a diferença na observação de critérios de usabilidade em interfaces.

Para avaliar se o aplicativo permite aos alunos aprender a respeito dos critérios de usabilidade, foi estabelecida a meta G2. Essa meta também possuía quatro perguntas (Q2.1 a Q2.4) que receberam respostas positivas após a aplicação da avaliação.

A pergunta Q2.1 avaliou se os alunos conseguiam reconhecer as medidas que comumente são utilizadas para avaliar a usabilidade de uma solução computacional. Essa pergunta foi feita diretamente aos alunos, antes e depois de eles utilizarem o aplicativo, e para a qual foi apresentado um conjunto com oito alternativas como resposta, entre os quais as que se buscava verificar (*eficiência*, *eficácia* e *satisfação*), além da possibilidade das opções “*Não sei*” e “*Outros*” (ver Apêndice A). Mais de uma alternativa poderia ser marcada como resposta por aluno.

Antes do uso do aplicativo, a alternativa mais marcada foi *adequação*, com 22 alunos escolhendo essa opção (71%), seguida por *funcionalidades*, escolhida por 19 alunos (61,3%). Somente depois vieram as alternativas *eficiência*, *eficácia* e *satisfação*, com 13 (41,9%), 15 (48,4%) e 16 (51,6%) alunos respectivamente. 5 alunos marcaram a alternativa “*Não sei*” (16,1%). Após o uso do aplicativo, as alternativas mais marcadas foram *eficiência*, *eficácia* e *satisfação*, com 24 (77,4%), 27 (87,1%) e 25 (80,6%) alunos respectivamente. Em seguida vieram as opções *adequação*, escolhida por 18 alunos (58,1%), e *funcionalidades*, escolhida por 14 alunos (45,2%). 2 alunos marcaram “*Não sei*” (6,5%).

Esses valores mostram que, após o uso do aplicativo, mais alunos passaram a reconhecer a *eficiência*, a *eficácia* e a *satisfação* como medidas de usabilidade. No entanto, a experiência não foi suficiente para mostrar aos alunos que outras medidas não se aplicam ao tema, apesar do trabalho não ter esse objetivo. Uma discussão posterior, a ser feita entre o professor e os alunos, pode ajustar

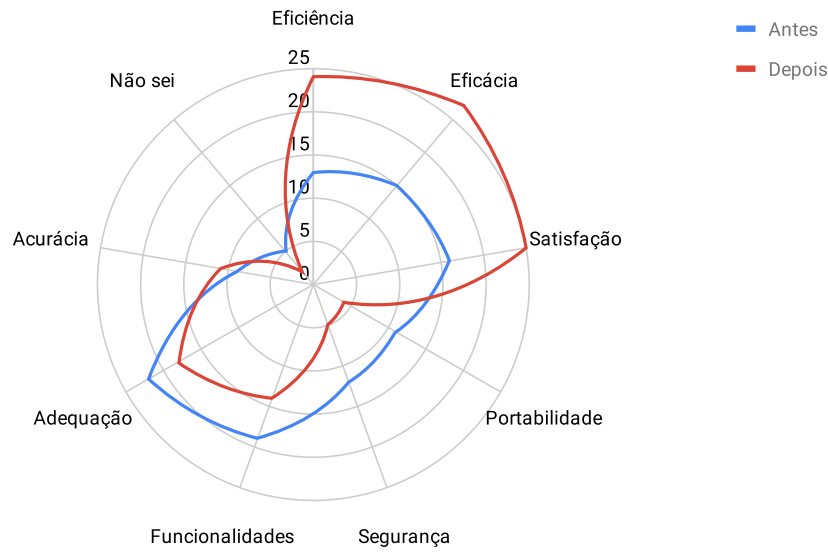


Figura 29 – Medidas de usabilidade escolhidas antes e depois do uso do aplicativo

a eliminar possíveis confusões.

Para verificar se os alunos perceberam como medir a *eficiência*, a *eficácia* e a *satisfação* foi elaborada a pergunta Q2.2 e suas métricas. Anteriormente ao uso da aplicação, 10 alunos (32,3%) responderam que a *eficiência* poderia ser medida através do tempo gasto e da quantidade de cliques necessários para concluir uma tarefa e 9 alunos (29%) responderam que era através da quantidade de tarefas concluídas com sucesso. Já quanto à *eficácia*, 10 alunos (32,3%) responderam que ela poderia ser medida através do tempo gasto e da quantidade de cliques necessários para concluir uma tarefa e 11 alunos (35,5%) responderam que era através da quantidade de tarefas concluídas com sucesso. Esse equilíbrio entre as respostas indica que os termos *eficiência* e *eficácia* não parecem estar muito claros para eles. Quanto à *satisfação*, 16 alunos (51,6%) responderam que era medida pela aplicação de um questionário subjetivo.

Após a utilização da aplicação, 19 alunos (61,3%) responderam corretamente como medir a *eficiência* e a *satisfação* e 16 alunos (51,6%) acertaram como medir a *eficácia*, representando um significativo aumento. No entanto, uma possível confusão entre os termos *eficiência* e *eficácia* parece não ter se dissipado completamente, pois o número de alunos que inverteram a forma de medição desses conceitos manteve-se em 10 (31,3%). Novamente, entende-se que a discussão do tema em sala de aula pode contribuir para resolver o problema.

A verificação do reconhecimento dos critérios de usabilidade por parte dos alunos foi feita utilizando a pergunta Q2.3. No formulário preenchido após o uso do aplicativo, primeiramente perguntou-se ao alunos se eles conseguiam reconhecer os diversos aspectos que impactam na facilidade de uso de uma aplicação. Nas respostas, graduadas na escala Likert, 26 alunos (83,9%)

atribuíram valores 4 ou 5. Em seguida, os alunos descreveram, em uma questão aberta, algum aspecto que impactou negativamente na sua experiência de uso. Apesar de não citarem os critérios explicitamente, 25 respostas puderam ser mapeadas a um ou mais critérios de usabilidade (ver Apêndice B). A Figura 30 traz a quantidade de vezes que cada critério foi citado.

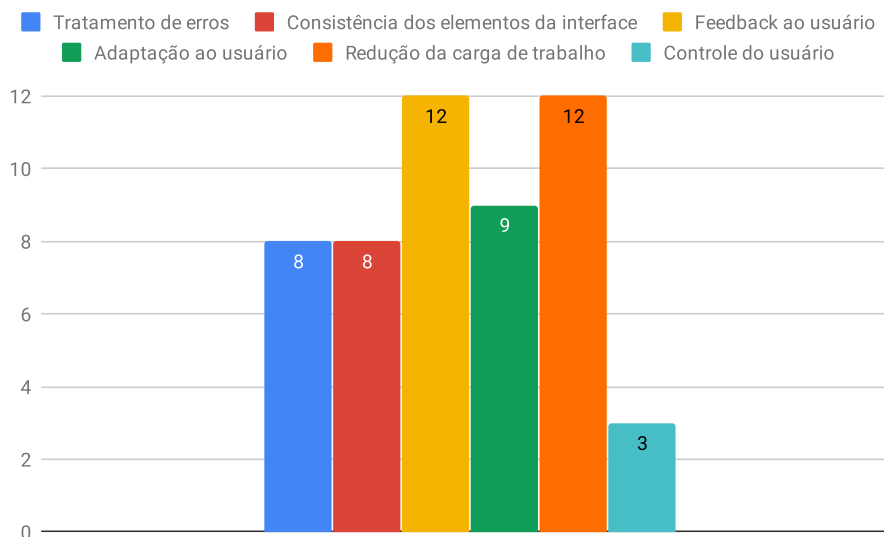


Figura 30 – Critérios de usabilidade reconhecidos

Alguns exemplos de respostas são as seguintes:

- “Formulário sem validação (*Tratamento de erros: prevenção e correção*) e quando mandei sem todas as infos ele apagou tudo (*Redução da carga de trabalho*).”
- “Quando ocorreu um erro no preenchimento do formulário, mas não me foi dito qual foi o erro e onde (*Feedback ao usuário*).”
- “Ao preencher o formulário de participação ou o de certificado o ‘Tab’ não pulava ao próximo input. . . (*Adaptação ao usuário*)”

Mesmo não citando os aspectos explicitamente, os alunos perceberam alguns dos critérios violados e como isso impactou negativamente em sua experiência. Dessa forma, há muitos elementos que podem ser discutidos posteriormente entre o professor e os alunos.

Por último, a pergunta Q2.4 buscava apreender o quanto a experiência com o aplicativo foi interessante para os alunos. No formulário respondido após a interação, todas utilizando a escala Likert de 5 pontos, 29 alunos (93,5%) responderam 4 ou 5 à questão “Eu percebi a importância da usabilidade para a qualidade de um produto de software”, 25 (80,6%) à questão “A experiência

com o aplicativo vai contribuir para meu desempenho na vida profissional” e 25 (80,6%) à questão “Eu recomendaria este aplicativo para meus colegas.”

Esse valores são significativos para mostrar que os alunos que participaram desta avaliação perceberam o propósito da aplicação e entenderam a importância que a observação de critérios de usabilidade tem para melhorar a interação entre o usuário e a máquina.

Com essas respostas, entende-se que houve algum tipo de aprendizado por parte dos alunos sobre o tema *usabilidade* e a percepção de sua importância. Acredita-se também que o papel do professor para ajudar a esclarecer e dirimir dúvidas não é prejudicado, mas ao contrário, ampliado com o uso da ferramenta.

6 Considerações Finais

Neste trabalho, foram analisadas algumas definições de usabilidade, assim como sua importância. Além disso, foi realizado um estudo comparativo entre alguns dos principais conjuntos de critérios de usabilidade que são amplamente utilizados em um método de avaliação conhecido como Avaliação Heurística. Como resultado, percebeu-se que alguns desses critérios são comumente propostos pelos principais pesquisadores da área. Esses foram os principais critérios empregados na construção da aplicação de software deste trabalho.

Em seguida, foi conduzido um mapeamento sistemático para que se pudesse ter uma visão mais ampla a respeito das recentes pesquisas na área de ensino de IHC, mais especificamente na utilização de metodologias e ferramentas de suporte ao ensino de usabilidade. Como resultado, esse estudo verificou que abordagens ativas são largamente empregadas em aulas de IHC e que, comparativamente com professores outras áreas, como de Engenharia de Software, por exemplo, os professores de IHC não contam com grande variedade de ferramentas desenvolvidas com o intuito de servir como suporte pedagógico. Essa última constatação suporta a relevância deste trabalho. Essa etapa do trabalho resultou ainda na confecção do artigo intitulado “*Let’s talk about teaching HCI*”, que será apresentado no *21st International Conference on Human-Computer Interaction*, em Orlando, EUA, no próximo mês de julho.

Como resultado do estudo, houve o desenvolvimento de uma aplicação que proporcionou a alunos da disciplina Engenharia de Software do curso de graduação em Sistemas de Informação que nunca tiveram contato com o tema a possibilidade de identificar os principais aspectos da usabilidade e descobrir como eles podem ser medidos em uma interação para realizar tarefas específicas. Além disso, eles puderam reconhecer alguns critérios de usabilidade e perceber como as medidas são alteradas quando alguns desses critérios são adequadamente aplicados ou não no design de interfaces. Na avaliação do aplicação, mais de 90% desses alunos declararam terem percebido a importância da usabilidade para a qualidade de um produto de software e mais de 80% afirmaram que a experiência com o aplicativo vai contribuir para seu desempenho na vida profissional.

6.1 Trabalhos Futuros

Como sugestões de trabalhos futuros a serem desenvolvidos a partir dos resultados aqui apresentados pode ser mencionado o seguinte:

- Aplicação de testes de usabilidade para que se possa acompanhar de forma sistemática a utilização da aplicação pelo usuário e propor melhorias nas interfaces;

- Análise estatística dos resultados que possam servir de base a uma interpretação mais precisa dos dados coletados pela aplicação;
- Desenvolvimento de novos estudos de caso, com novas tarefas, de modo a diversificar a experiência dos alunos, ou envolvendo protótipos de diferentes níveis de fidelidade;
- Internacionalização da aplicação para que possa ser amplamente utilizada;
- Separação da aplicação UsabilidadeZero dos estudos de caso para que ela possa ser utilizada para análise de outras aplicações.

Referências

- ACM/IEEE-CS Joint Task Force on Computing Curricula. *Computer Science Curricula 2013*. [S.l.], 2013. Disponível em: <<http://dx.doi.org/10.1145/2534860>>.
- AHRAM, T.; KARWOWSKI, W.; SCHMORROW, D. (Ed.). *Proc. of the 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences*, v. 3. Las Vegas, EUA: Elsevier, 2015. 6660 p.
- ALNUAIM, A.; CALEB-SOLLY, P.; PERRY, C. Enhancing Student Learning of Human-Computer Interaction Using a Contextual Mobile Application. In: IEEE. *Proc. of 2016 SAI Computing Conference (SAI)*. Londres, Reino Unido, 2016. p. 952–959.
- ANDRE, T. (Ed.). *Advances in Human Factors in Training, Education, and Learning Sciences*, v. 596. Los Angeles, EUA: Springer International Publishing, 2018.
- BANGOR, A.; KORTUM, P. T.; MILLER, J. T. An Empirical Evaluation of the System Usability Scale. *Intl. Journal of Human-Computer Interaction*, Taylor & Francis, v. 24, n. 6, p. 574–594, 2008.
- BASIL, V. R.; CALDIERA, G.; ROMBACH, H. D. *Goal, Question Metric Paradigm*. EUA: Wiley, 1994.
- BASTIEN, J. C.; SCAPIN, D. L. *Ergonomic criteria for the evaluation of human-computer interfaces*. [S.l.], 1993. 79 p. Disponível em: <<https://hal.inria.fr/inria-00070012>>.
- BATTISTELLA, P. E. *ENgAGED: Um processo de desenvolvimento de jogos para ensino em computação*. 401 f. Tese (Doutorado em Ciências da Computação) — Universidade Federal de Santa Catarina, Florianópolis, 2016.
- BENITTI, F. B. V.; SOMMARIVA, L. Evaluation of a Game Used to Teach Usability to Undergraduate Students in Computer Science. *Journal of Usability Studies*, Usability Professionals' Association, Bloomington, IL, v. 11, n. 1, p. 21–39, nov. 2015. ISSN 1931-3357. Disponível em: <<http://dl.acm.org/citation.cfm?id=2870660.2870663>>.
- BORGES, T. S.; ALENCAR, G. Metodologias Ativas na Promoção da Formação Crítica do Estudante: O Uso das Metodologias Ativas como Recurso Didático na Formação Crítica do Estudante do Ensino Superior. *Cairu em Revista*, Fundação Visconde de Cairu, Salvador, n. 4, p. 119–143, jul/ago 2014. ISSN 22377719. Disponível em: <<http://www.cairu.br/revista/artigos4.html>>.
- BRABRAND, C. *Palestra: How to make sure your students learn what you want them to*. Brasil: Universidade Federal de Pernambuco, 2010.
- CHONG, F. Teaching usability in a technical communication classroom: Developing competencies to user-test and communicate with an international audience. In: IEEE. *Proc. of 2012 IEEE International Professional Communication Conference*. Orlando, EUA, 2012. p. 1–4. ISSN 2158-091X.

- CHONG, F. Implementing Usability Testing in Introductory Technical Communication Service Courses: Results and Lessons From a Local Study. *IEEE Transactions on Professional Communication*, PP, n. 99, p. 1–10, 2017. ISSN 0361-1434.
- CYBIS, W. d. A.; BETIOL, A. H.; FAUST, R. *Ergonomia e Usabilidade - Conhecimentos, Métodos e Aplicações*. 2. ed. São Paulo: Novatec Editora, 2010.
- DJAJALAKSANA, Y. M. *A National Survey of Instructional Strategies Used to Teach Information Systems Courses: An Exploratory Investigation*. 210 f. Tese (Doctor of Philosophy) — University of South Florida, Tampa, 2011.
- FENG, G.; LUO, B. An Experience of Teaching HCI to Undergraduate Software Engineering Students. In: *IEEE. Proc. of 2012 IEEE 25th Conference on Software Engineering Education and Training*. Nanjung, China, 2012. p. 125–129. ISSN 1093-0175.
- FERRAZ, A. P. do C. M.; BELHOT, R. V. Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. *Gestão e Produção*, scielo, São Carlos, v. 17, n. 2, p. 421–431, 00 2010. ISSN 0104-530X.
- FERREIRA, B. M. et al. UsabiliCity: Um Jogo de Apoio ao Ensino de Propriedades de Usabilidade de Software Através de Analogias. In: *Anais do Simpósio Brasileiro de Informática na Educação*. Dourados, Brasil: Sociedade Brasileira de Computação – SBC, 2014. p. 1273–1282. ISSN 2316-6533.
- GIL, A. C. *Métodos e Técnicas de Pesquisa Social*. 6. ed. São Paulo: Atlas, 2008. ISBN 978-85-224-5142-5.
- HUNDHAUSEN, C. D. et al. Woz pro: A pen-based low fidelity prototyping environment to support wizard of oz studies. In: *Proc. of CHI '07 Extended Abstracts on Human Factors in Computing Systems*. Nova Iorque, EUA: ACM, 2007. p. 2453–2458. ISBN 978-1-59593-642-4.
- HUNDHAUSEN, C. D.; FAIRBROTHER, D.; PETRE, M. An Empirical Study of the “Prototype Walkthrough”: A Studio-Based Activity for HCI Education. *ACM Transactions on Computer-Human Interactions*, ACM, Nova Iorque, EUA, v. 19, n. 4, p. 26:1–26:36, dez. 2012. ISSN 1073-0516.
- ISO. *ISO 9241-11:2018 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability*. [S.l.], 2018.
- JEON, M. What to Teach in HCI?: How to Educate HCI Students to Envision the Future of Human Being, not the Future of Technology? In: SAGE PUBLICATIONS SAGE CA. *Proc. of the Human Factors and Ergonomics Society Annual Meeting*. Los Angeles, EUA, 2015. v. 59, n. 1, p. 362–366.
- KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Durham, UK, 2007.
- LAZEM, S. A case study for sensitising egyptian engineering students to user-experience in technology design. In: ACM. *Proc. of the 7th Annual Symposium on Computing for Development*. Nairobi, Quênia, 2016. p. 12.

- LESTER, C. Y. Advancing the Multidisciplinary Nature of Human Computer Interaction in a Newly Developed Undergraduate Course. In: IARIA. *Prof. of the First International Conference on Advances in Computer-Human Interaction*. Sainte-Luce, Martinica, 2008. p. 177–182.
- MARCUS, A. (Ed.). *Proc. of the 15th International Conference on Human-Computer Interaction: Second international conference on design, user experience, and usability: Health, learning, playing, cultural, and cross-cultural user experience*, v. 8013. Las Vegas, EUA: Springer, 2013.
- MOLICH, R.; NIELSEN, J. Improving a Human-Computer Dialogue. *Communications of the ACM*, ACM, Nova Iorque, EUA, v. 33, n. 3, p. 338–348, mar 1990. ISSN 0001-0782.
- NIELSEN, J. Enhancing the Explanatory Power of Usability Heuristics. In: *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. Nova Iorque, EUA: ACM, 1994. (CHI '94), p. 152–158. ISBN 0-89791-650-6.
- NIELSEN, J. *10 Usability Heuristics for User Interface Design*. 1995. Disponível em: <<https://www.nngroup.com/articles/ten-usability-heuristics/>>. Acesso em: 25 de julho de 2017.
- NIELSEN, J. *Success Rate: The Simplest Usability Metric*. 2001. Disponível em: <<https://www.nngroup.com/articles/success-rate-the-simplest-usability-metric/>>. Acesso em: 14 de agosto de 2017.
- NIELSEN, J. *Usability 101: Introduction to Usability*. 2012. Disponível em: <<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>>. Acesso em: 1º de agosto de 2017.
- NIELSEN, J. *User Satisfaction vs. Performance Metrics*. 2012. Disponível em: <<https://www.nngroup.com/articles/satisfaction-vs-performance-metrics/>>. Acesso em: 11 de agosto de 2017.
- OLIVEIRA, K. Marçal de et al. Teaching Task Analysis for User Interface Design: Lessons Learned from Three Pilot Studies. In: ACM. *Proc. of the 27th Conference on L'Interaction Homme-Machine*. Toulouse, France, 2015. (IHM '15), p. 31:1–31:6. ISBN 978-1-4503-3844-8.
- OR-BACH, R. Design and Implementation of an HCI course for MIS students – Some lessons. *Issues in Informing Science and Information Technology*, Directory of Open Access Journals, v. 12, p. 153–163, 2015.
- PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, Elsevier, v. 64, p. 1–18, 2015. ISSN 0950-5849.
- POLACK-WAHL, J. A. Teaching HCI in Software Engineering. In: IEEE. *Proc. of the 34th Annual Frontiers in Education (FIE 2004)*. Savannah, EUA, 2004. p. F1C–31–6 Vol. 2. ISSN 0190-5848.
- PREECE, J.; ROGERS, Y.; SHARP, H. *Design de Interação: Além da interação homem-computador*. 1. ed. Porto Alegre: Bookman, 2005.
- PRESSMAN, R. S. *Engenharia de Software*. 6. ed. Porto Alegre: AMGH, 2010.
- PRINCE, M. Does active learning work? A review of the research. *Journal of engineering education*, Wiley Online Library, v. 93, n. 3, p. 223–231, 2004.

- RUSU, C.; RUSU, V. Human-Computer Interaction from theory to practice. In: *Anales del VIII Congreso Iberoamericano de Informática Educativa*. San José, Costa Rica: [s.n.], 2006.
- SANTOSO, H. B.; SARI, E. Transforming undergraduate HCI course in Indonesia: a preliminary study. In: *ACM. Proc. of the Asia Pacific HCI and UX Design Symposium*. Melbourne, Austrália, 2015. p. 55–59.
- SBC. *Currículo de Referência da SBC para Cursos de Graduação em Bacharelado em Ciência da Computação e Engenharia de Computação*. [S.l.], 2005. Disponível em: <<http://www.sbc.org.br/documentos-da-sbc/summary/131-curriculos-de-referencia/760-curriculo-de-referencia-cc-ec-versao2005>>. Acesso em: 17 de outubro de 2017.
- SCIALDONE, M.; CONNOLLY, A. J. A Creative Approach to Devising Non-Technical, Meaningful Exercises in Human-Computer Interaction Undergraduate Education. In: *ISCAP. Proc. of the EDSIG Conference*. Las Vegas, EUA, 2016. v. 2473, p. 3857.
- SHNEIDERMAN, B. *The Eight Golden Rules of Interface Design*. 2016. Disponível em: <<https://www.cs.umd.edu/users/ben/goldenrules.html>>. Acesso em: 10 de agosto de 2017.
- SHNEIDERMAN, B.; PLAISANT, C. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 5. ed. EUA: Addison-Wesley Publishing Company, 2009. ISBN 9780321537355.
- SILVA, E. L. da; MENEZES, E. M. *Metodologia da Pesquisa e Elaboração de Dissertação*. 3. ed. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001.
- SOMMARIVA, L. W. *UsabilityGame: Jogo Simulador para apoio ao ensino de Usabilidade*. 210 f. Dissertação (Mestrado em Computação Aplicada) — Universidade do Vale do Itajaí, São José, 2012.
- SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011. ISBN 978-85-7936-108-1.
- TOPI, H. et al. Curriculum guidelines for undergraduate degree programs in information systems. *ACM/AIS task force*, 2010.
- URQUIZA-FUENTES, J.; PAREDES-VELASCO, M. Investigating the effect of realistic projects on students' motivation, the case of human-computer interaction course. *Computers in Human Behavior*, Elsevier, v. 72, p. 692–700, 2017.
- WANGENHEIM, C. G. von; SHULL, F. To game or not to game? *IEEE Software*, IEEE, v. 26, n. 2, p. 92–94, 2009.
- WANGENHEIM, C. G. von; WANGENHEIM, A. von. *Ensinando Computação com Jogos*. 1. ed. Florianópolis: Bookess Editora, 2012. ISBN 9788580454628.
- WAZLAWICK, R. S. *Análise e Projeto de Sistemas de Informação Orientados a Objetos*. 2. ed. Rio de Janeiro: Elsevier, 2011. ISBN 978-85-352-3916-4.
- WIERINGA, R. et al. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, Springer, v. 11, n. 1, p. 102–107, 2006.

YAMAMOTO, S. (Ed.). *Proc. of the 15th International Conference on Human-Computer Interaction: Human interface and the management of information*, v. 8018. Las Vegas, EUA: Springer, 2013.

YI, Y. et al. *Interest-Based Learning for Teaching a Human-Computer Interaction Course: Media and Cognition Course*. 2017.

ZAHARIAS, P.; BELK, M.; SAMARAS, G. Employing virtual worlds for HCI education: A problem-based learning approach. In: *Proc. of CHI '12 Extended Abstracts on Human Factors in Computing Systems*. Nova Iorque, EUA: ACM, 2012. (CHI EA '12), p. 317–326. ISBN 978-1-4503-1016-1.

ZAPHIRIS, P.; IOANNOU, A. *Proc. of the 16th International Conference on Human-Computer Interaction: Learning and Collaboration Technologies: Designing and Developing Novel Learning Experiences, Part I*. 1. ed. Creta, Grécia: Springer, 2014. v. 8523.

ZAPHIRIS, P.; IOANNOU, A. *Proc. of the 16th International Conference on Human-Computer Interaction: Learning and Collaboration Technologies: Designing and Developing Novel Learning Experiences, Part I*. 1. ed. Creta, Grécia: Springer, 2014. v. 8524.

ZAPHIRIS, P.; IOANNOU, A. *Proc. of the 17th International Conference on Human-Computer Interaction: Learning and Collaboration Technologies*. 1. ed. Los Angeles, EUA: Springer, 2015. v. 9192.

ZAPHIRIS, P.; IOANNOU, A. *Proc. of the 18th International Conference on Human-Computer Interaction: Learning and Collaboration Technologies*. 1. ed. Toronto, Canadá: Springer, 2016. v. 9753.

ZAPHIRIS, P.; IOANNOU, A. *Proc. of the 19th International Conference on Human-Computer Interaction: Learning and Collaboration Technologies. Novel Learning Ecosystems, Part I*. 1. ed. Vancouver, Canadá: Springer, 2017. v. 10295.

ZAPHIRIS, P.; IOANNOU, A. *Proc. of the 19th International Conference on Human-Computer Interaction: Learning and Collaboration Technologies. Novel Learning Ecosystems, Part II*. 1. ed. Vancouver: Springer, 2017. v. 10296.

APÊNDICE A – Formulário de Avaliação

Avaliação do aplicativo UsabilidadeZero

Esta pesquisa objetiva avaliar o uso do aplicativo UsabilidadeZero, que é parte do Projeto de Conclusão de Curso do aluno Adriano Luiz de Souza Lima e orientado pela Prof. Dra. Fabiane Barreto Vavassori Benitti.

***Obrigatório**

1. Desejo participar *

Marcar apenas uma oval.

Sim

Não Comece este formulário novamente.

Informe o seu email

2. *

Identificação do perfil

3. Você já estudou ou tem algum conhecimento prévio sobre Engenharia de Usabilidade?

*

Marque uma ou mais opções.

Marque todas que se aplicam.

Sim. Eu fiz um curso/disciplina sobre usabilidade.

Sim. Eu trabalhei/trabalho com testes de usabilidade.

Sim. Eu estudei/estudo sobre o tema por interesse próprio.

Não.

4. Você já participou do teste de usabilidade de alguma solução computacional? *

Marcar apenas uma oval.

Sim, como participante/usuário do sistema.

Sim, como membro da equipe de avaliação.

Não.

Quanto ao seu conhecimento sobre usabilidade, responda:

5. Quais aspectos são comumente verificados para avaliar a usabilidade de uma solução computacional? *

Marque todas que se aplicam.

- Satisfação
- Eficiência
- Segurança
- Não sei
- Funcionalidades
- Acurácia
- Portabilidade
- Eficácia
- Adequação
- Outro: _____

Quanto ao seu conhecimento sobre usabilidade, responda:

6. Em um teste de usabilidade, como pode ser medida a EFICIÊNCIA de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Quanto ao seu conhecimento sobre usabilidade, responda:

7. Em um teste de usabilidade, como pode ser medida a SATISFAÇÃO de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Quanto ao seu conhecimento sobre usabilidade, responda:

8. Em um teste de usabilidade, como pode ser medida a ADEQUAÇÃO de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Quanto ao seu conhecimento sobre usabilidade, responda:

9. Em um teste de usabilidade, como pode ser medida a ACURÁCIA de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Quanto ao seu conhecimento sobre usabilidade, responda:

10. Em um teste de usabilidade, como pode ser medida a EFICÁCIA de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Quanto ao seu conhecimento sobre usabilidade, responda:

11. Em um teste de usabilidade, como podem ser medidas as FUNCIONALIDADES de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Teste do aplicativo

Agora você deve parar de responder a este questionário e iniciar o uso do aplicativo UsabilidadeZero, no endereço <http://limitless-wave-90100.herokuapp.com/> antes de continuar a responder a este questionário.

Lembre-se: Você NÃO DEVE FECHAR ESTA PÁGINA.

Continuar questionário

Agora que você utilizou o aplicativo e pôde verificar a diferença entre ambientes com usabilidade diferentes, continue a responder este questionário.

Quanto ao seu conhecimento sobre usabilidade, responda:

12. Quais aspectos são comumente verificados para avaliar a usabilidade de uma solução computacional? *

Marque todas que se aplicam.

- Segurança
- Acurácia
- Satisfação
- Adequação
- Eficiência
- Funcionalidades
- Portabilidade
- Eficácia
- Não sei
- Outro: _____

Quanto ao seu conhecimento sobre usabilidade, responda:

13. Em um teste de usabilidade, como pode ser medida a EFICIÊNCIA de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Quanto ao seu conhecimento sobre usabilidade, responda:

14. Em um teste de usabilidade, como pode ser medida a SATISFAÇÃO de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Quanto ao seu conhecimento sobre usabilidade, responda:

15. Em um teste de usabilidade, como pode ser medida a ADEQUAÇÃO de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Quanto ao seu conhecimento sobre usabilidade, responda:

16. Em um teste de usabilidade, como pode ser medida a ACURÁCIA de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Quanto ao seu conhecimento sobre usabilidade, responda:

17. Em um teste de usabilidade, como pode ser medida a EFICÁCIA de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Quanto ao seu conhecimento sobre usabilidade, responda:

18. Em um teste de usabilidade, como podem ser medidas as FUNCIONALIDADES de uma solução computacional? *

Marcar apenas uma oval.

- Através da quantidade de tarefas concluídas com sucesso.
- Através do tamanho de sua documentação.
- Através do tempo gasto e da quantidade de cliques para concluir uma tarefa.
- Aplicando um questionário subjetivo ao usuário.
- Analisando o uso das cores na interface.
- Através do tempo de processamento de suas instruções
- Não é uma medida relacionada a usabilidade.
- Não sei.

Sua experiência em relação ao uso do aplicativo

19. Eu consegui reconhecer diversos aspectos que impactam na facilidade de uso de uma aplicação. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo fortemente

20. Descreva detalhadamente um aspecto que impactou negativamente na sua experiência de uso. *

Em relação ao uso do aplicativo, avalie as sentenças abaixo:

21. Foi fácil entender o que o aplicativo propunha e começar a utilizá-lo. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo fortemente

22. Eu percebi a importância da usabilidade para a qualidade de um produto de software. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo fortemente

23. A experiência com o aplicativo vai contribuir para meu desempenho na vida profissional. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo fortemente

24. Eu recomendaria este aplicativo para meus colegas. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo fortemente

Comentário final:

25. Comente sobre a sua experiência com o aplicativo. Você pode sugerir melhorias, fazer um elogio ou críticas.

APÊNDICE B – Mapeamento dos Critérios de Usabilidade

DESCREVA DETALHADAMENTE UM ASPECTO QUE IMPACTOU NEGATIVAMENTE NA SUA EXPERIÊNCIA DE USO.		Tratamento de erros	Consistência dos elementos da interface	Feedback ao usuário	Adaptação ao usuário	Redução da carga de trabalho	Controle do usuário
1	a div dos botões de desistir e concluído ocupavam mais da metade da tela, era quase impossível usar o site						
2	As cores (no caso 0). Além do uso de apenas siglas (novamente no caso 0).				x	x	
3	Cores ruins, informações perdidas, formulários que não falam o que há de errado.		x	x	x		
4	Péssima navegabilidade.		x		x		
5	As siglas dificultam muito a legibilidade da aplicação					x	
6	falha na organização dos tópicos		x				
7	Abreviações, Cores Fortes,				x	x	
8	No primeiro caso algumas funções básicas como tab não estavam disponíveis. Formulário sem validação e quando mandei sem todas as infos ele apagou tudo. Difícil encontrar as informações também. Não consegui imprimir o certificado em ambos os casos	x		x	x	x	x
9	Quando ocorreu um erro no preenchimento do formulário, mas não me foi dito qual foi o erro e onde.	x		x			
10	Formulário sem indicação de formato e validação dos valores inseridos.	x		x			
11	formulario						
12	Nomes do eventos em abreviações, sem botão de voltar pra Home, falta tab order, cor na interface para guiar o uso.		x		x	x	
13	Disposição dos dados na tela		x				
14	Inconsistência de informações, falta das mesmas		x	x			
15	A ferramenta foi boa para constatar as dificuldades que um usuário tem com um sistema cuja usabilidade não segue princípios, porém sinto que a mesma não foi tão eficiente para transmitir os conceitos teóricos em si.						
16	falta de textos explicativos			x			
17	não poder testar valores quaisquer nos campos pré determinados	x		x		x	
18	O portal principal não especifica nem sequer quais os simpósios listados com sucesso, sendo apenas siglas nem nenhuma outra informação que confunde o usuário que busca um evento em específico. Além de mais de um Simpósio que utiliza a mesma sigla.	x		x			

	DESCREVA DETALHADAMENTE UM ASPECTO QUE IMPACTOU NEGATIVAMENTE NA SUA EXPERIÊNCIA DE USO.	Tratamento de erros	Consistência dos elementos da interface	Feedback ao usuário	Adaptação ao usuário	Redução da carga de trabalho	Controle do usuário
19	o caso 0 era horrível, não vou comentar; um ponto que não gostei no caso 1 foi ter que selecionar novamente a opção de pagamento, após ter que preencher um campo obrigatório que foi esquecido. Foi assim: 1) esqueci de preencher um campo; 2) seleccionei a forma de pagamento; 3) cliquei para finalizar a inscrição; 4) fui informado para preencher todos os campos; 5) preenchi o campo que faltava; 5) cliquei para finalizar a inscrição; 6) NOVAMENTE fui informado a preencher todos os campos; pois a forma de pagamento selecionada anteriormente não havia sido salva	x				x	
20	A experiência de se navegar em um sistema com poucas informações, como era o caso do primeiro sistema apresentado. Ele apresentava apenas a abreviação dos eventos e nada mais que caracterizasse ele em sí. Nele eu tive dificuldade para encontrar o evento que eu queria fazer as operações em cima e, só isso, já me deixou insatisfeito o suficiente para ser um motivo de para de usar a aplicação.			x			
21	Eficiência						
22	A cor forte, difícil de ficar olhando. Cansa o olho muito rápido. (caso 0)				x		
23	Não consegui encontrar onde se tira o comprovante de inscrição de um usuário.						x
24	Siglas dos simpósios no primeiro caso e troca de lugares após clicar no voltar do navegador, funcionalidades escondidas ou "pequenas"		x			x	
25	A subjetividade na interação com o sistema e suas funcionalidades.						
26	Falta de retorno quanto à campos preenchidos incorretamente, apenas apagando todos os campos.	x		x		x	
27	Ao preencher o formulário de participação ou o de certificado o "Tab" não pulava ao próximo input, o que impactou de forma extremamente negativa, me obrigando ao uso do mouse em uma situação completamente desnecessária.				x	x	
28	Tentar perceber inicialmente o objectivo do questionário						
29	A falta de informações claras sobre o que se trata tal página ou tal botão, a falta de um guia da página por tópicos			x	x		
30	quando se deparava com uma entrada incorreta, o sistema simplesmente removia todas as informações e exibia uma mensagem de erro genérica, que inviabilizava a identificação e correção da informação incorreta.	x		x		x	x
31	Sistema não era nada intuitivo, siglas atrapalhavam. Não havia conexão entre as ações que eu tomava.		x			x	

APÊNDICE C – Artigo

Author Proof



Let's Talk About Tools and Approaches for Teaching HCI

Adriano Luiz de Souza Lima^(B) and Fabiane Barreto Vavassori Benitti

Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil
{adriano.lima, fabiane.benitti}@ufsc.br

Abstract. Human-computer interaction (HCI) is an important knowledge field in the program of most computer-related majors, but not many studies on HCI teaching at undergraduate level can be found. This paper presents a systematic mapping study carried out in order to get an overview of HCI classes at undergraduate level, with the objective of investigating how HCI is being taught and what tools are being used to support the process. The study selected 17 papers to analyze the teaching approach being applied and the supporting tools, if any, being used to help teaching HCI. As a result, most of the papers mentioned or described an active learning approach and only two specific tools to support HCI teaching have been identified, which comes as a future work opportunity.

[AQ1](#)

[AQ2](#)

Keywords: Human-computer interaction · Computing education · HCI education

1 Introduction

In the last 20 years, the development of technology considerably changed the way people interact to each other. Before that time, mobile phones had very little use besides making calls, fewer people had Internet access and most of online communication was done by email. Today people communicate with each other by sending and receiving messages through various forms such as text, audio, images, video or symbols, synchronous or asynchronously.

During this period, education has always kept its doors open to the use of new technologies such as simulators, just to mention a single example. In spite of that, communication between teachers and students in class seem not to have taken full advantage of the possibilities the advances of technology have brought. Most of the classes are still based on oral lectures, using few visual aids, where teachers just speak and students ask their questions when they think something is not clear [1]. This type of class hardly draws students' attention [2] and they, as a result, usually have a low rate of content retention [3].

Students commonly feel little involved in the communication process of more traditional teaching methods since, in general, they allow them have a more passive posture towards their teachers that, on the other hand, must be very

2 A. L. de Souza Lima and F. B. V. Benitti

active in their classes in order to keep students motivated [3]. Also, those methods do not usually offer the students many opportunities to apply that recently acquired knowledge to real life situations, leading to a superficial learning level that could be represented by the initial categories of the cognitive domain of Bloom's taxonomy [4], for instance.

Receiving considerable attention over the past several years, active learning has found many advocates among faculty looking for alternatives to traditional teaching, even though there may still exist some that regard it as just an educational fad [5].

Human-Computer Interaction (HCI) is an important area in computer related degree programs, such as computer science [6], information systems [7] and software engineering [8]. Despite its importance, there seems to be a lack of literature on practice-level issues about its implementation in the classroom [9]. Likewise, technological tools specifically designed to aid HCI teaching are practically nonexistent [10]. Considering this scenario, the present paper conducts a systematic mapping study that aims at checking how HCI is being taught and if any tools to support the process are being used.

The remaining of this paper is organized as follows: Sect. 2 presents related work. Section 3 describes the systematic mapping with the presentation of its results. Section 4 discusses the implication of the results, answering the research questions and presenting the threats to the validity of the study. Conclusions and comments on future work are described in Sect. 5.

2 Background

The importance of HCI can be stated when looking at the Curriculum Guidelines for Undergraduate Degree Programs for Computer Science [6], for Information Systems [7] and for Software Engineering [8]. Computer science students are expected to have at least 6.4 lecture hours of HCI core topics and have a choice of some other elective ones [6]. When majoring in information systems, the students might take Introduction to HCI as an elective course, though it brings significant coverage to the application developer and user interface designer career tracks, whereas the program for software engineering [8] considers that it is essential that students have 10 hours of HCI.

By looking at those programs, and understanding the role HCI plays on interface designing, one might think that a considerable number of software developers have at least been initiated on the matter. Edwards et al. [11], however, argues that this might be true, but teachers are failing at HCI education. As an evidence of his position, he points out the abundance of poorly designed interfaces and great number of papers published in the HCI literature that do not offer much more than criticisms of interface designs. He concludes that there is a vast scope for improvement in HCI teaching.

Battistella and Wangenheim [12] carried out a systematic review of the literature to understand what kind of games are available for teaching computing in higher education. They encountered a considerable number of 107 games,

indicating that there exists a trend to game-based learning also in computing education. However, only one ranked match in the HCI area was found. The game called "3DAR Lego Game" provides a tool to improve spatial ability for a wide range of ages [13].

In 2012, Sommariva [14] conducted a systematic mapping study to search for games or simulators specifically developed to support usability teaching, and also to understand what usability topics were being taught and how. His study found no games developed to support usability teaching. He later developed and proposed a game to help teach usability engineering life cycle [10]. When it comes to usability teaching, his study focused on the activity proposed by the teacher, without mentioning the underlying teaching approach. Later, in 2014, Ferreira et al. [15] presented a game to teach Jacob Nielsen's heuristic evaluation. The game makes use of analogies to reinforce the heuristics understanding, building a story to motivate the students during their learning process.

Nevertheless, systematic reviews (or systematic mappings) that analyzed approaches and computational solutions specifically for teaching HCI were not identified. Therefore, this article proposes: "Let's talk about tools and approaches for teaching HCI".

3 Systematic Mapping

This work presents a systematic mapping study carried out to get an overview of HCI classes at the undergraduate level. A systematic mapping is usually used to investigate a wider research area than a systematic literature review when little or no evidence on a topic is known [16]. The interest here is to get an indication of the quantity of that evidence and classify it for further studies.

3.1 Planning and Conducting the Mapping

The phase of planning the mapping consists of developing a review protocol which defines the methods to undertake a specific systematic review. Having a well defined protocol reduces the possibility that this review can be driven by research expectations.

Review Objective and Research Questions. The main objective of this mapping is to identify how HCI is being taught to undergraduate students. In order to guide this analysis towards its achievement, it is important to know what are the teaching approaches that are mostly used in class and what tools, if any, are used to support it. With this in mind, three research questions are proposed:

- RQ1 What are the main approaches used to teach HCI to undergraduate students?
- RQ2 What are the available technological tools specifically developed to support HCI teaching?
- RQ3 How have the approaches or technologies used to teach HCI been evaluated?

4 A. L. de Souza Lima and F. B. V. Benitti

Search Strategy. An automated search strategy was used in three different scientific databases: ACM Digital Library¹, IEEE Xplore Digital Library², Google Scholar³. These results were complemented with manual search on selected conferences:

- International Conference on Applied Human Factors and Ergonomics and the Affiliated Conferences, AHFE 2015 [17] and AHFE 2017 [18];⁴
- HCI International 2013–2017 [19–25];⁵

The search string used is divided into three parts:

- Part1 This part of the string used expressions relating HCI and teaching. By using only the expression “HCI”, without relating it to teaching, most of the studies returned by the search discussed on the applications in the area, but outside the education context. If on one hand the search was restricted, on the other hand, it was extended by adding term “usability”. The choice for this term is supported in [26], that states “Usability and HCI are becoming core aspects of the system development process to improve and enhance system facilities and to satisfy users’ needs and necessities.”
- Part2 Here the string was limited to search for papers that dealt with teaching undergraduates.
- Part3 The last part had the intention to find among those papers the ones that explicitly mentioned how their teaching was conducted, using some kind of tool, approach or both.

In this way, the string used for the search was:

(“teaching hci” OR “hci teaching” OR “hci education” OR “teaching usability” OR “usability teaching” OR “usability education”) AND undergraduate AND (software OR game OR simulation OR tool OR environment OR methodology).

To complement the automated procedure, a snowballing search, backward and forward, was conducted following the guidelines in [27]. All the works selected from the automated search were used as seeds.

¹ dl.acm.org.

² ieeexplore.ieee.org.

³ scholar.google.com.

⁴ When searching in the proceedings of International Conference on Applied Human Factors and Ergonomics (AHFE), only the conferences held in 2015 and 2017 were considered since those were the only years when the affiliated conference “Advances in Human Factors in Training, Education, and Learning Sciences” took place. AHFE was not held in 2013.

⁵ The search in HCI International conference, in 2013, aimed at the affiliated conference “Design, User Experience, and Usability (Part II: Health, Learning, Playing, Cultural, and Cross-Cultural User Experience; and Part III: Information and Interaction for Learning, Culture, Collaboration and Business)”, whereas in the years 2014–2017, it aimed at the affiliated conference “Learning and Collaboration Technologies”. The conference was not held in 2012.

Inclusion and Exclusion Criteria. Some inclusion and exclusion criteria were defined for selecting papers for the final review.

The mapping should only include:

- Peer-reviewed publications appearing in journals, conferences, and workshops;
- Publications written in English;
- Papers published from 2012 to 2017;
- Papers that discussed about or presented a methodology or an approach for teaching HCI on undergraduate level;
- Papers describing computational solutions to support HCI teaching.

The following works should be excluded from the mapping:

- Presentation documents, such as PowerPoint slides, and short/extended abstract papers;
- Papers that did not relate to undergraduate HCI courses;
- Papers that, although related to HCI teaching undergraduates, discussed about other aspects of the course or program, such as curriculum proposal, for instance;
- Papers that described computational solutions applying concepts related to HCI area, but that did not focus on teaching those concepts.

Data Extraction and Data Synthesis Strategies. The mapping was conducted from December 2017 to February 2018 with the execution of the protocol, and resulted in the selection of 21 papers for data extraction. However, during the data extraction process it was observed that in three selected studies there was no consistent information for extraction, that is, the information was not sufficient to respond to the RQs. So these 3 papers were excluded. The number of retrieved, examined and selected papers from each resource is summarized in Table 1. The selection process is illustrated in Fig. 1.

Data were extracted from each selected paper following 15 previously defined information items (Table 2).

Results. Table 3 summarizes the data extracted for the 17 articles selected considering some information from groups 1 and 2 of Table 2:

- The Context column indicates where the research described in the paper was conducted, i.e., which course, with its major in parenthesis;
- The In-Text Topics column presents those HCI topics the authors mentioned they had worked with the students during their research;
- The HCI Topics column classifies the topics surveyed by considering an HCI course in a curriculum guideline for undergraduate degree programs [7];
- The methodologies/ approaches used to teach HCI are in the column Approach;
- The last column, Tools, indicates the tools used during the teaching process that were mentioned in the text.

6 A. L. de Souza Lima and F. B. V. Benitti

Table 1. Search result

Resource	# Papers found	# Papers examined	# Papers selected
ACM	6	6	2
IEEE	24	24	4
Google Scholar	353	100	9
AHFE	938	938	0
HCI International	439	439	0
Backward snowballing	421	421	2
Forward snowballing	51	51	0
Total selected	2,232	1,979	17

Table 2. Data extracted from each primary study selected

Group	Information item
Group 1: Publication identification	II1. Publication ID II2. Publication title II3. Year of publication II4. Authors' name II5. Publication source
Group 2: Context reported in the publication	II6. Context of research (course/ major) II7. Specific topics being taught II8. HCI topics [7] II9. Approach applied II10. Justification for use of approach II11. Technological tool used II12. Use of the tool
Group 3: Evaluation described in the publication	II13. Research type classification [28] II14. Evaluation II15. Sample

During the period researched, the publication of selected studies seemed to have been fairly regular, with 3 to 7 papers being published a year. The exception was in the 2013–2014 period, when no papers were published (Fig. 2).

From the 17 selected papers, 12 (70.5%) conducted their research in a HCI course, whether mandatory or elective, whereas 5 of them (29.5%) had their researches in some other courses (technical communication [TC], system design [SD], interaction design [ID] or winter school), where some topic of HCI was taught.

Let's Talk About Tools and Approaches for Teaching HCI 7

Table 3. Selected papers summary

ID	Ref. Context*	In-Text Topics	HCI Topics [7]	Approach	Tools
1	[30] HCI (CS)	Usability lifecycle; Requirements analysis; Prototyping; Heuristic evaluation	Development; Evaluation methods	Serious game	Computer game (UsabilityGame)
2	[31] SD (CS)	Task analysis	Special HCI issues	-	ConcurTaskTrees Environment (CTTE)
3	[32] HCI (CS)	User interface design project	Development	Studio-based learning	Simple art supply; Prototyping tools (WOZ Pro); HTML
4	[33] HCI (CS)	User-centered design process with emphasis on interface design	User-centered design	Problem-based learning	Virtual world platform (OpenSimulator)
5	[34] HCI (CS, IS)	Usability testing	Evaluation methods	Active learning	Student Centered e-Learning Environment (SCELE)
6	[35] winter school (Eng)	Interaction design, usability, user experience, prototyping, data collection and analysis	User-centered design; Development	Action research approach	-
7	[36] HCI (SE)	Basic Foundation, Requirement Analysis, Interaction Design, Usability Evaluation	Principles in HCI design; Development; Evaluation methods	Project development	-
8	[37] HCI (not informed)	Prototyping; Evaluation; Usability testing	Development; Evaluation methods	Blended learning model	sLearn
9	[9] TC (S, T, Eng, M)	Usability testing	Evaluation methods	User-centered approach	-
10	[38] TC (Eng, CS, B)	Usability testing	Evaluation methods	Service-learning projects	-
11	[39] HCI classes (P, D, Eng)	Core HCI design cycles	Development	Scaffolding approach	-
12	[40] HCI (CS, ISc)	Principles of design and usability, with an emphasis on the human-side of interactions	User-centered design	Hands-on experiential activities	-
13	[41] HCI (MIS)	Interface design with emphasis on the users	User-centered design	Studio-based learning	-
14	[42] HCI (CS)	Single-modal and multiple modal cognitive theories	Principles in HCI design	Interest-Based Learning	-
15	[43] HCI (CS)	Requirement collection and analysis; Interface design; Interface evaluation	Development; Evaluation methods	Realistic projects approach	-
16	[44] HCI (CS)	HCI4D (HCI for Development)	Development	Student-centered, hands-on approach	-
17	[45] ID (not informed)	Design thinking	Development	Design thinking approach; studio-based teaching	-

* Legend of abbreviations: [CS] Computer Science; [M] Mathematics; [HCI] Human-Computer Interaction; [IS] Information Systems; [B] Business; [SD] System Design; [Eng] Engineering; [P] Psychology; [TC] Technical Communication; [SE] Software Engineering; [D] Design; [S] Science; [ISc] Information Science; [T] Technology; [MIS] Management Information Systems; [ID] Interaction design

4 Discussion

With the data extracted from the selected papers it is possible now to answer the research questions.

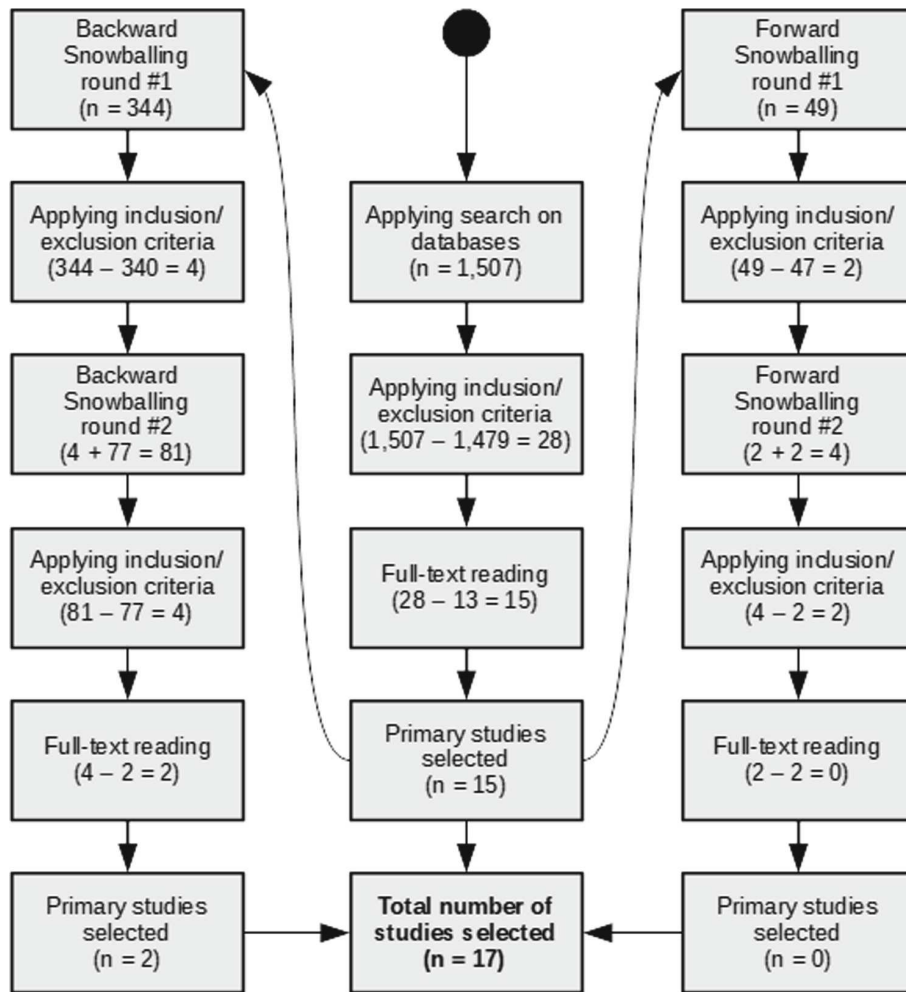


Fig. 1. Process of selecting primary studies

R Q1 What are the main approaches used to teach HCI to undergraduate students?

From the data extracted here, a wide variety of teaching approaches or techniques can be identified (Fig. 2).

Most articles mentioned the use of a specific approach (blue dots in the graph of Fig. 2). However, some articles did not cite a specific approach, and in this case, they were ranked from the analysis of the text (red dots in the graph of Fig. 2):

- Experiential activities: experiential learning is a guided process of questioning, investigating, reflecting, and conceptualizing based on direct experiences.

Let's Talk About Tools and Approaches for Teaching HCI

9

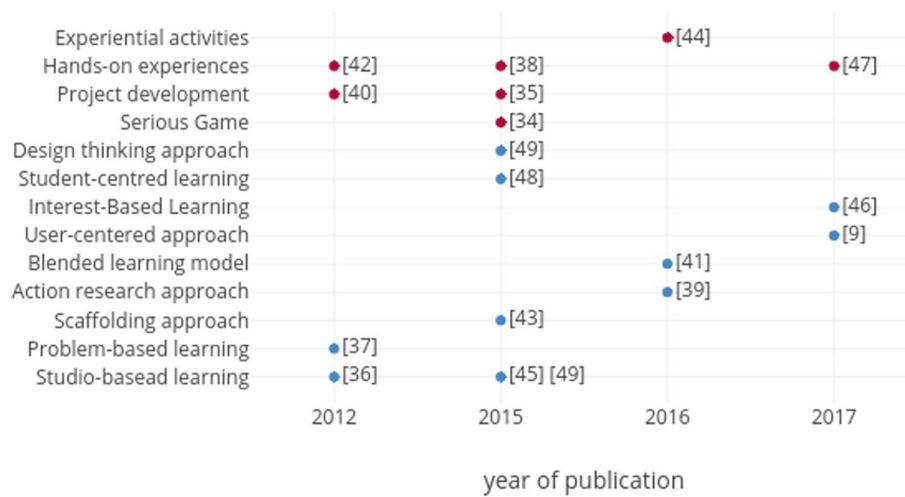


Fig. 2. Approaches used in selected studies

In this learning process, the learner is actively engaged, has freedom to choose, and directly experiences the consequences of their actions [45].

- Hands-on experiences: development of projects with real users, that is, real-world cases.
- Project development: development of projects without considering real users.
- Serious game: use of serious game (educational game) to support teaching.

Among the articles that did not mention a specific approach, the teaching of HCI through the development of group projects using as case study real needs, that is, real-world cases [33, 37, 42] can be highlight as the most recurrent.

Nine different approaches were cited in the selected studies. For Zaharias et al. [32] problem-based learning develops “experiential and social learning and calls for an active rather than passive approach to learning that leads to the development of critical thinking skills”. Jeon [38], on the other hand, defends that although a problem-solving approach may not work for novices (i.e., undergraduates), “they are still encouraged to be involved in the overall design process, but they start with a focus on part of the problem with well-defined guidance based on the scaffolding approach.” Culén [44] proposes the use of design thinking that “stands firmly on three main pillars: empathy with users and human centeredness, rapid prototyping to generate large number of alternatives in order to solve the right problem rather than a problem right, and last, but not least, their synthesis leading to best viable and feasible solutions that incorporate desired values” [44].

Chong [9] mentions the use of user-centered approach, whereas, Taylor et al. [43] prefer student-centered learning which are research-led, problem-based and flexible, where students can focus on the topics and content delivery methods which are most interesting and useful to them. Alnuain et al. [36] apply blended

10 A. L. de Souza Lima and F. B. V. Benitti

learning model “a learning approach that contains different types of education techniques and technologies. This learning model aims to provide more effective education experiences by combining features and functions of well-known learning and teaching techniques” [46].

Studio-based learning was the most cited approach [31, 40, 44]. Adapted from architecture and art education, as well as from collaborative problem-solving pedagogies, studio-based learning has shown great promise for computing education [47]. Key elements of studio based learning include exploring multiple solutions to a problem, justifying the choice of one solution, and being subject to, as well as providing, peer reviews [47].

Yang et al. [41] propose a new approach, called interest-based learning, in this approach “students are guided to organize teams by same personal hobbies or specialties, e.g. music or sports, and conduct research on HCI in these familiar and interested topic.”

It can be observed from this analysis that the teaching of HCI to undergraduates has used some kind of active learning approach, considered here as “any instructional method that engages students in the learning process” [5]. The popularity of active learning among HCI teachers seems to be clear when assuming that this approach tries to involve the students in a way that is closer to the professional environment. Most of the importance of HCI for the IT industry is related to practical activities such as interface prototyping and development or usability testing, for instance. The approaches identified on the selected papers try to actively involve the students at doing something rather than passively receiving information. For Hundhausen et al. [31], giving the students a more central participation “is akin to the situation of an expert designer presenting a user interface design to a design team in a real-world company.”

RQ2 What are the available technological tools specifically developed to support HCI teaching?

Although HCI is closely related to computers and some other technological devices, the use of tools specifically developed for HCI teaching does not seem to be very widespread. From the selected papers, only 6 mentioned some kind of tool during the process and, among them, only 2 described tools that were specifically designed for the purpose.

One of those tools is a serious game, called UsabilityGame [29], that offers the students the opportunity to practice the usability lifecycle by addressing requirement analysis, prototyping and heuristic evaluation. When playing the game, the students have to select the correct set of requirements for each scenario presented, develop a prototype to be evaluated by the teacher and evaluate real-world interfaces by choosing one of Nielsen’s heuristics [48] that would correctly improve the detected usability issue. The main role of the teacher, during the game, is to set up the game according to the student’s learning objectives, to conduct the evaluation of the prototypes, and monitor the performance of the class.

The other tool is WOZ Pro (Wizard of Oz Prototyper) [31], a low-fidelity prototyping environment that was developed to support prototyping creation

and wizard of oz testing. Although it was conceived to be used by college students who are first learning about interaction design, this tool could also be used outside the educational context [49].

A few other tools such as an environment for editing and analysis of task models (CTTE - ConcurTaskTrees Environment [30]), an e-learning environment (SCELE - Student Centered e-Learning Environment [33]), a virtual world platform (OpenSimulator [32]) and even simple art supply [49] were mentioned to be used during the teaching process and they can all be as valuable as the computer-based tools described above. However, HCI teachers seem not to have many choices when picking out some supporting tool that could meet their educational purposes.

R Q3 How have the approaches or technologies used to teach HCI been evaluated?

The selected studies were also classified following the research type classification proposed by [28]. According to this study, most of the primary studies (76%) were classified as experience papers, presenting the authors' experiences in class. The importance of these papers rely in the relevance of the lessons learned by the author from that experience [50]. Three other studies (18%) were classified as evaluation research, with the proposal of a novel piece of knowledge in HCI education. At last, 1 study (6%) was classified as an opinion paper (see Table4).

Despite the importance of experience papers, that contribute with experiences of in-practice approaches, revealing their results when applied in real contexts, the number of studies evaluating new developments in HCI education does not seem to be very large. This high percentage of experience studies on HCI education seems to indicate that researchers are mostly concentrated on applying their efforts to understanding the area, producing data to pave the road of further developments.

When evaluating the result of the applied approaches: (i) a simple survey was used in 7 studies [9,34-37,41,44] without statistical tests; (ii) 4 studies applied statistical analysis to evaluate their results (Wilcoxon test [29], Pearson Product-Moment Correlation Coefficient [40], Kruskal-Wallis, Mann-Whitney [42], chi-squared test and ANOVA [31]); (iii) interview [9,40] and structured interviews were used by [32]; (iv) observation was the technique used by [32,36,40]; and (v) content analysis (video) used by [31].

The size of the sample used by the researchers ranged from 8 [44] to approximately 1,150 [35] students participating in the process. When analyzing the mean and the median numbers (159 and 40, respectively), they are considerably higher than those of the studies published at CHI 2014 [51].

4.1 Threats to Validity of the Mapping

One of the major problems with systematic reviews is finding all the relevant primary studies (evidence). In this case, three search strategies were used to

12 A. L. de Souza Lima and F. B. V. Benitti

Table 4. Research type classification [28]

Research type	Papers
Experience papers	[9, 30, 32–37, 39–41, 43, 44]
Evaluation research	[29, 31, 42]
Opinion paper	[38]

ensure that the largest number of related studies would be found: (i) Automatic search in 3 different databases; (ii) Manual search in specific proceedings of two conferences of HCI area not indexed on the searched databases; and (iii) backward and forward snowballing.

In addition, our search strings were designed to find the maximum number of relevant papers. Nonetheless, it is possible that they have missed studies that used different terminology to describe any part of the string.

The search resulted in some papers that, in spite of having their titles and abstracts in English, were actually written in other languages. Those papers were excluded from the selection, but, due to language limitations, it was not possible to assess whether they had relevant information or not.

Readers must consider that a systematic review is by definition limited by the search date, the electronic sources and key terms used in the search. Therefore, it is possible that other papers may be included in a future replication of the study. The results are limited by the previous features, and by the evolution of the HCI education area itself.

5 Conclusions

Although HCI teaching is present in many computer-related programs, it seems that HCI teaching in undergraduate level has not been drawing enough attention from researchers, what is evidenced by the small number of selected papers. This conclusion gets patent when looking at the dates of the publications and realizing that for two years in a row there were no selected publications on the theme. The lack of selected papers from the proceedings of prestigious conferences shows that the road is still open for research in HCI education.

Most of the selected papers presented researches on either development or evaluation methods teaching. Nevertheless, the teaching of some topics, such as Relevance of HCI and Devices, was not mentioned. The lack of research on those topics does not mean they are less important than others, it simply means that there is a field ahead to be explored.

Active teaching approaches as a whole seem to be the main option when teaching HCI to undergraduate students. The present study did not try to categorize the multitude of approaches or evaluate its results when applied to this area of knowledge, because the prevalence of one approach over the others could not be stated in small number of primary studies found. This categorization might eventually become an object of future work.

HCI teachers have very few options when choosing a supporting tool exclusively developed for their classes. The same does not happen in other areas of knowledge on computer field. On a paper of 2009, for example, Wangenheim and Shull [52] found 16 games used in software engineering education, most of them computer-based, but also card or board games. Although the paper concludes that more research in the area is necessary, it is clear that there were a higher number of educational games on software engineering 10 years ago than there are of educational tools for HCI today. Future work on these tools would bring great benefit for HCI teachers and students.

Finally, it is noted that few empirical studies have been conducted to verify the results obtained by the different teaching approaches and tools used. Thus, this paper suggests that more empirical studies be performed with sufficient rigor to improve the body of evidence in the HCI education field.

References

1. Djajalaksana, Y.M.: A National Survey of Instructional Strategies Used to Teach Information Systems Courses: An Exploratory Investigation. Doctor of Philosophy, University of South Florida, Tampa (2011)
2. von Wangenheim, C.G., von Wangenheim, A.: *Ensinando Computação com Jogos*, 1st edn. Bookess Editora, Florianópolis (2012)
3. Brabrand, C.: *How to make sure your students learn what you want them to*. Palestra (2010)
4. do Carmo Marchetti Ferraz, A.P., Belhot, R.V.: Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. *Gestão e Produção* 17(2), 421-431 (2010)
5. Prince, M.: Does active learning work? A review of the research. *J. Eng. Educ.* 93(3), 223-231 (2004)
6. Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society: *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, 999133. ACM, New York (2013)
7. Topi, H., et al.: *Curriculum Guidelines for Undergraduate Degree Programs in Information Systems*. ACM/AIS Task Force (2010)
8. The Joint Task Force on Computing Curricula: *Curriculum guidelines for undergraduate degree programs in software engineering*. Technical report, ACM/IEEE, New York (2015)
9. Chong, F.: Implementing usability testing in introductory technical communication service courses: results and lessons from a local study. *IEEE Trans. Prof. Commun.* P P (99), 1-10 (2017)
10. Sommariva, L.W.: *Usabilitygame*. Mestrado em computação aplicada, Universidade do Vale do Itajaí, São José (2012)
11. Edwards, A., Wright, P., Petrie, H.: HCI education: we are failing-why. In: *Proceedings of HCI Educators Workshop* (2006)

14. A. L. de Souza Lima and F. B. V. Benitti
12. Battistella, P.E., von Wangenheim, C.G.: Games for teaching computing in higher education - a systematic review. *IEEE Technol. Eng. Educ. (ITEE)* 1, 8-30 (2016)
13. Do, T.V., Lee, J.-W.: A multiple-level 3D-LEGO game in augmented reality for improving spatial ability. In: Jacko, J.A. (ed.) *HCI 2009*. LNCS, vol. 5613, pp. 296-303. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02583-9_33
14. Benitti, F.B.V., Sommariva, L.: Investigando o ensino de ihc no contexto da computação: o que e como é ensinado? In: *Anais do III Workshop sobre Ensino de IHC* (2012)
15. Ferreira, B.M., Rivero, L., Lopes, A., Marques, A.B., Conte, T.: UsabiliCity: Um Jogo de Apoio ao Ensino de Propriedades de Usabilidade de Software Através de Analogias. In: *Anais do Simpósio Brasileiro de Informática na Educação*, Dourados, MT, Sociedade Brasileira de Computação - SBC, pp. 1273-1282 (2014)
16. Kitchenham, B., Charters, S.: Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical report EBSE 2007-001, Keele University and Durham University Joint Report, Durham, UK (2007)
17. Ahram, T., Karwowski, W., Schmorow, D.: 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences. *Procedia Manufacturing*, vol. 3, p. 6660 (2015)
18. Andre, T.: *Advances in Human Factors in Training, Education, and Learning Sciences*. AISC, vol. 596, 1st edn. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-60018-5>
19. Marcus, A.: *Design, User Experience, and Usability: Health, Learning, Playing, Cultural, and Cross-Cultural User Experience*. LNCS, vol. 8013, 1st edn. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-39241-2>
20. Yamamoto, S.: *Human Interface and the Management of Information: Information and Interaction for Learning, Culture, Collaboration and Business*, LNCS, vol. 8018, 1st edn. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-39226-9>
21. Zaphiris, P., Ioannou, A.: *Learning and Collaboration Technologies: Designing and Developing Novel Learning Experiences*. LNCS, vol. 8523, 1st edn. Springer, Cham (2014). <https://doi.org/10.1007/978-3-319-07482-5>
22. Zaphiris, P., Ioannou, A.: *Learning and Collaboration Technologies: Technology-Rich Environments for Learning and Collaboration*. LNCS, vol. 8524, 1st edn. Springer, Cham (2014). <https://doi.org/10.1007/978-3-319-07485-6>
23. Zaphiris, P., Ioannou, A.: *Learning and Collaboration Technologies*. LNCS, vol. 9192, 1st edn. Springer, Cham (2015). <https://doi.org/10.1007/978-3-319-20609-7>
24. Zaphiris, P., Ioannou, A.: *Learning and Collaboration Technologies*. LNCS, vol. 9753, 1st edn. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-39483-1>
25. Zaphiris, P., Ioannou, A.: *Learning and Collaboration Technologies: Novel Learning Ecosystems*. LNCS, vol. 10295, 1st edn. Springer, Cham (2017). <https://doi.org/10.1007/978-3-319-58509-3>
26. Issa, T., Isaias, P.: *Sustainable Design*, pp. 19-36. Springer, London (2015). https://doi.org/10.1007/978-1-4471-6753-2_2
27. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE 2014, pp. 38:1-38:10. ACM, New York (2014)
28. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: an update. *Inf. Softw. Technol.* 64, 1-18 (2015)

29. Benitti, F.B.V., Sommariva, L.: Evaluation of a game used to teach usability to undergraduate students in computer science. *J. Usability Stud.* 11(1), 21–39 (2015)
30. Marçal de Oliveira, K., Girard, P., Gonçalves, T.G., Lepreux, S., Kolski, C.: Teaching task analysis for user interface design: lessons learned from three pilot studies. In: *Proceedings of the 27th Conference on L'Interaction Homme-Machine. IHM 2015*, pp. 31:1–31:6. ACM, New York (2015)
31. Hundhausen, C.D., Fairbrother, D., Pêtre, M.: An empirical study of the “prototype walkthrough”: a studio-based activity for HCI education. *ACM Trans. Comput.-Hum. Interact.* 19(4), 26:1–26:36 (2012)
32. Zaharias, P., Belk, M., Samaras, G.: Employing virtual worlds for HCI education: a problem-based learning approach. In: *CHI 2012 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2012*, pp. 317–326. ACM, New York (2012)
33. Santoso, H.B., Sari, E.: Transforming undergraduate HCI course in Indonesia: a preliminary study. In: *Proceedings of the Asia Pacific HCI and UX Design Symposium*, pp. 55–59. ACM (2015)
34. Lazem, S.: A case study for sensitising Egyptian engineering students to user-experience in technology design. In: *Proceedings of the 7th Annual Symposium on Computing for Development*, p. 12. ACM (2016)
35. Feng, G., Luo, B.: An experience of teaching HCI to undergraduate software engineering students. In: *2012 IEEE 25th Conference on Software Engineering Education and Training*, pp. 125–129, April 2012
36. Alnuaim, A., Caleb-Solly, P., Perry, C.: Enhancing student learning of human-computer interaction using a contextual mobile application. In: *2016 SAI Computing Conference (SAI)*, pp. 952–959, July 2016
37. Chong, F.: Teaching usability in a technical communication classroom: developing competencies to user-test and communicate with an international audience. In: *2012 IEEE International Professional Communication Conference*, pp. 1–4, October 2012
38. Jeon, M.: What to teach in HCI?: How to educate HCI students to envision the future of human being, not the future of technology? *Proc. Hum. Factors Ergon. Soc. Ann. Meeting* 59(1), 362–366 (2015)
39. Scialdone, M., Connolly, A.J.: A creative approach to devising non-technical, meaningful exercises in human-computer interaction undergraduate education. In: *Proceedings of the EDSIG Conference* ISSN, vol. 2473, p. 3857 (2016)
40. Or-Bach, R.: Design and implementation of an HCI course for MIS students-some lessons. *Issues Inf. Sci. Inf. Technol.* 12(unknown), 153–163 (2015)
41. Yi, Y., Shengjin, W., Jiasong, S., Xian, Z.: Interest-based learning for teaching a human-computer interaction course: media and cognition course. In: *Proceedings of the 2017 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS 2017)* (2017)
42. Urquiza-Fuentes, J., Paredes-Velasco, M.: Investigating the effect of realistic projects on students' motivation, the case of human-computer interaction course. *Comput. Hum. Behav.* 72, 692–700 (2017)
43. Taylor, J.L., Tsimeris, J., Zhu, X., Stevenson, D., Gedeon, T.: Observations from teaching HCI to Chinese students in Australia. In: *Proceedings of the ASEAN CHI Symposium 2015*, pp. 31–35. ACM, New York (2015)
44. Culén, A.L.: HCI education: innovation, creativity and design thinking. In: *Proceedings of the The Eighth International Conference on Advances in Computer-Human Interactions, ACHI 2015* (2015)
45. Obrenović, Z.: Rethinking hci education: teaching interactive computing concepts based on the experiential learning paradigm. *ACM Interact.* 19(3), 66–70 (2012)

- 16 A. L. de Souza Lima and F. B. V. Benitti
46. Köse, U.: A blended learning model supported with web 2.0 technologies. *Procedia - Soc. Behav. Sci.* 2(2), 2794–2802 (2010). *Innovation and Creativity in Education*
 47. Hendrix, D., Myneni, L., Narayanan, H., Ross, M.: Implementing studio-based learning in CS2. In: *Proceedings of the 41st ACM Technical Symposium on Computer Science Education, SIGCSE 2010*, pp. 505–509. ACM, New York (2010)
 48. Nielsen, J.: 10 usability heuristics for user interface design (1995). <https://www.nngroup.com/articles/ten-usability-heuristics/>. Accessed 25 July 2017
 49. Hundhausen, C.D., Balkar, A., Nuur, M., Trent, S.: WOZ pro: a pen-based low fidelity prototyping environment to support wizard of OZ studies. In: *CHI 2007 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2007*, pp. 2453–2458. ACM, New York (2007)
 50. Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requir. Eng.* 11(1), 102–107 (2006)
 51. Caine, K.: Local standards for sample size at CHI. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 981–992. ACM (2016)
 52. von Wangenheim, C.G., Shull, F.: To game or not to game? *IEEE Softw.* 26(2), 92–94 (2009)

APÊNDICE D – Código-Fonte

```
/*--- HomeController.java ---*/

package controllers;

import javax.inject.Inject;

import autenticadores.UsuarioAutenticado;
import models.Estudo;
import models.Usuario;
import play.data.Form;
import play.data.FormFactory;
import play.mvc.Controller;
import play.mvc.Result;
import play.mvc.Security.Authenticated;
import views.html.instrucaoEstudo;
import views.html.login;
import views.html.sobre;

public class HomeController extends Controller {

    private Form<Usuario> usuarioForm;
    private Form<Estudo> estudoForm;

    @Inject
    public HomeController(FormFactory formFactory) {

        this.usuarioForm = formFactory.form(Usuario.class);
        this.estudoForm = formFactory.form(Estudo.class);
    }

    public Result index() {

        return ok(login.render(usuarioForm));
    }
}
```

```
@Authenticated(UsuarioAutenticado.class)
public Result mostrarSobre() {

    return ok(sobre.render(
        "UsabilidadeZero",
        play.core.PlayVersion.current()));
}

@Authenticated(UsuarioAutenticado.class)
public Result mostrarInstrucoes() {

    Estudo form = estudoForm.bindFromRequest().get();

    return ok(instrucaoEstudo.render(form.isTipo()));
}
}
```

```
/*--- ControladorUsuario.java ---*/
```

```
package controllers;

import java.util.List;
import java.util.Optional;

import javax.inject.Inject;

import autenticadores.UsuarioAutenticado;
import daos.EstudoDAO;
import daos.TokenCadastroDAO;
import daos.UsuarioDAO;
import models.EmailCadastro;
import models.Estudo;
import models.TokenCadastro;
import models.TokenSistema;
import models.Usuario;
import play.api.libs.mailer.MailerClient;
import play.data.Form;
import play.data.FormFactory;
import play.mvc.Controller;
```

```
import play.mvc.Result;
import play.mvc.Security.Authenticated;
import validadores.Encriptador;
import validadores.Validador;
import views.html.login;
import views.html.painel;
import views.html.resultados;

public class ControladorUsuario extends Controller {

    @Inject
    private Validador validador;
    @Inject
    private MailerClient mailer;
    @Inject
    private UsuarioDAO usuarioDAO;
    @Inject
    private EstudoDAO estudoDAO;

    private Form<Usuario> usuarioForm;
    private Form<Estudo> estudoForm;

    public static final String AUTH = "auth";

    @Inject
    public ControladorUsuario(FormFactory formFactory) {

        this.usuarioForm = formFactory.form(Usuario.class);
        this.estudoForm = formFactory.form(Estudo.class);
    }

    public Result entrar() {

        Form<Usuario> form = usuarioForm.bindFromRequest();

        if(validador.temErros(form)) {

            flash("danger", "Foram identificados problemas no cadastro!");
            return badRequest(login.render(usuarioForm));
        }
    }
}
```

```
}

Usuario usuario = form.get();
String email = usuario.getEmail();

String criptoSenha = Encriptador.crypt("senhaAutomatica");

Optional<Usuario> possivelUsuario = usuarioDAO.comEmail(email);

if(possivelUsuario.isPresent()) {

    Usuario usuarioCadastrado = possivelUsuario.get();
    String senhaCadastrada = usuarioCadastrado.getSenha();

    if(senhaCadastrada.equals(criptoSenha)) {

        if(usuarioCadastrado.isVerificado()) {

            session(AUTH, usuarioCadastrado.getToken().getCodigo());
            flash("success", "Login realizado com sucesso!");

            return redirect(routes.ControladorUsuario.mostrarPainel());

        } else {

            flash("danger", "Usuario nao confirmado!");
            return redirect(routes.HomeController.index());

        }

    } else {

        String msg = String.format("A senha digitada nao partence ao usuario
            %s. Tente novamente.", email);

        flash("danger", msg);
        return redirect(routes.HomeController.index());

    }

} else {
```

```
        usuario.setSenha(criptoSenha);
        usuario.save();

        TokenCadastro token = new TokenCadastro(usuario);
        token.save();

        confirmarCadastro(email, token.getCodigo());

        return redirect(routes.ControladorUsuario.mostrarPainel());
    }
}

public Result entrarComSenha() {

    Form<Usuario> form = usuarioForm.bindFromRequest();

    if(validador.temErros(form)) {

        flash("danger", "Foram identificados problemas no cadastro!");
        return badRequest(login.render(usuarioForm));
    }

    Usuario usuario = form.get();
    String email = usuario.getEmail();

    String criptoSenha = Encriptador.crypt(usuario.getSenha());

    Optional<Usuario> possivelUsuario = usuarioDAO.comEmail(email);

    if(possivelUsuario.isPresent()) {

        Usuario usuarioCadastrado = possivelUsuario.get();
        String senhaCadastrada = usuarioCadastrado.getSenha();

        if(senhaCadastrada.equals(criptoSenha)) {

            if(usuarioCadastrado.isVerificado()) {
```

```
        session(AUTH, usuarioCadastrado.getToken().getCodigo());
        flash("success", "Login realizado com sucesso!");

        return redirect(routes.ControladorUsuario.mostrarPainel());
    } else {

        flash("danger", "Usuario nao confirmado!");
        return redirect(routes.HomeController.index());
    }

} else {

    String msg = String.format("A senha digitada nao partence ao usuario
        %s. Tente novamente.", email);

    flash("danger", msg);
    return redirect(routes.HomeController.index());
}

} else {

    usuario.setSenha(criptoSenha);
    usuario.save();

    TokenCadastro token = new TokenCadastro(usuario);
    token.save();

    mailer.send(new EmailCadastro(token));

    String msg = String.format("Usuario cadastrado com sucesso! Verifique o
        email enviado para o endereco %s "
        + "para confirmacao do seu cadastro.", email);

    flash("success", msg);

}

return redirect(routes.HomeController.index());
```

```
}

public Result confirmarCadastro(String email, String codigo) {

    Optional<TokenCadastro> possivelToken = TokenCadastroDAO.comCodigo(codigo);
    Optional<Usuario> possivelUsuario = usuarioDAO.comEmail(email);

    if(possivelToken.isPresent() && possivelUsuario.isPresent()) {

        TokenCadastro tokenCadastro = possivelToken.get();
        Usuario usuario = possivelUsuario.get();

        if(tokenCadastro.getUsuario().equals(usuario)) {

            tokenCadastro.delete();
            usuario.setVerificado(true);

            TokenSistema tokenSistema = new TokenSistema(usuario);
            tokenSistema.save();

            usuario.setToken(tokenSistema);
            usuario.update();

            flash("success", "Seu usuario foi confirmado com sucesso!");

            session(AUTH, usuario.getToken().getCodigo());

            return redirect(routes.ControladorUsuario.mostrarPainel());
        }
    }

    flash("danger", "Algo deu errado ao tentar confirmar o seu cadastro!");

    return redirect(routes.HomeController.index());
}

@Authenticated(UsuarioAutenticado.class)
public Result mostrarPainel() {
```



```
List<Estudo> estudos = estudoDAO.mostraTodos();

String codigo = session(AUTH);
    Usuario usuario = usuarioDAO.comToken(codigo).get();

    return ok(painel.render(estudoForm, resultados.render(estudos)));
}

@Authenticated(UsuarioAutenticado.class)
public Result sair() {

    session().clear();
    flash("success", "Voce saiu do sistema!");

    return redirect(routes.HomeController.index());
}

}
```

```
/*--- ControladorEstudos.java ---*/
```

```
package controllers;

import java.util.Calendar;
import java.util.Collections;
import java.util.List;
import java.util.Optional;

import javax.inject.Inject;

import autenticadores.UsuarioAutenticado;
import daos.EstudoDAO;
import daos.EventoDAO;
import daos.TarefaDAO;
import daos.TokenSistemaDAO;
import daos.UsuarioDAO;
import models.Estudo;
import models.Evento;
import models.RelatorioEstudo;
```

```
import models.Sus;
import models.Tarefa;
import models.TokenSistema;
import models.Usuario;
import play.data.Form;
import play.data.FormFactory;
import play.mvc.Controller;
import play.mvc.Result;
import play.mvc.Security.Authenticated;
import views.html.estudo0portal;
import views.html.estudo1portal;
import views.html.instrucaoEstudo;
import views.html.instrucaoSus;
import views.html.relatorioFinal;
import views.html.relatorioParcial;
import views.html.relatorioTarefa;
import views.html.sus;
import views.html.tarefa;

public class ControladorEstudos extends Controller {

    @Inject
    private EventoDAO eventoDAO;
    @Inject
    private UsuarioDAO usuarioDAO;
    @Inject
    private EstudoDAO estudoDAO;
    @Inject
    private TarefaDAO tarefaDAO;
    @Inject
    private TokenSistemaDAO tokenSistemaDAO;

    private Form<Estudo> estudoForm;
    private Form<Tarefa> tarefaForm;
    private Form<Sus> susForm;

    public static final String AUTH = "auth";

    @Inject
```

```
public ControladorEstudos(FormFactory formFactory) {

    this.estudoForm = formFactory.form(Estudo.class);
    this.tarefaForm = formFactory.form(Tarefa.class);
    this.susForm = formFactory.form(Sus.class);
}

@Authenticated(UsuarioAutenticado.class)
public Result mostrarEstudo() {

    Estudo form = estudoForm.bindFromRequest().get();
    Estudo estudo = estudoDAO.comId(form.getId()).get();

    if (estudo.isTipo()) {

        Estudo estudoZero = estudo.getRelacionado();
        concluirEstudo(estudo);
        return ok(relatorioFinal.render(estudoZero.getRelatorio(),
            estudo.getRelatorio(), estudoForm));

    } else {
        return ok(instrucaoEstudo.render(true));
    }
}

@Authenticated(UsuarioAutenticado.class)
public Result iniciarEstudoZero() {

    Estudo estudo;

    Optional<Estudo> possivelEstudo = estudoDAO.comToken(session(AUTH));
    if (possivelEstudo.isPresent()) {

        estudo = possivelEstudo.get();

        if (estudo.getTarefas().size() > 0 || estudo.isTipo()) {

            concluirEstudo(estudo);
            estudo = criarNovoEstudo(false);
        }
    }
}
```

```
    }

} else {

    estudo = criarNovoEstudo(false);
}

    estudo.save();

    return ok(tarefa.render(estudo, estudoForm, 1));
}

@Authenticated(UsuarioAutenticado.class)
public Result iniciarEstudoUm() {

    Estudo estudo;

    Optional<Estudo> possivelEstudo = estudoDAO.comToken(session(AUTH));
    if (possivelEstudo.isPresent()) {

        estudo = possivelEstudo.get();

        if (!estudo.isTipo()) {

            Estudo estudoZero = estudo;
            concluirEstudo(estudo);

            estudo = criarNovoEstudo(true);
            estudo.setRelacionado(estudoZero);

            estudoZero.setRelacionado(estudo);
            estudoZero.update();
        }

    } else {

        estudo = criarNovoEstudo(true);
    }
}
```

```
        estudo.save();

        return ok(tarefa.render(estudo, estudoForm, 1));
    }

    @Authenticated(UsuarioAutenticado.class)
    public Result continuarEstudo() {

        Estudo estudo = estudoDAO.comToken(session(AUTH)).get();

        if (estudo.getTarefas().size() >= 3) return ok(instrucaoSus.render(estudo,
            susForm));

        int numTarefa = (estudo.getTarefas().size() + 1);

        return ok(tarefa.render(estudo, estudoForm, numTarefa));
    }

    @Authenticated(UsuarioAutenticado.class)
    public Result iniciarPesquisa() {

        Estudo estudo = estudoDAO.comToken(session(AUTH)).get();

        return ok(sus.render(estudo, susForm));
    }

    @Authenticated(UsuarioAutenticado.class)
    public void concluirEstudo(Estudo estudo) {

        if (estudo.getToken() != null) {

            if (estudo.getSus() != null) gerarRelatorio(estudo);

            Calendar calendario = Calendar.getInstance();
            estudo.setFim(calendario.getTime());
            estudo.setToken(null);
            estudo.update();
        }
    }
}
```

```
@Authenticated(UsuarioAutenticado.class)
private void gerarRelatorio(Estudo estudo) {

    RelatorioEstudo relatorio = new RelatorioEstudo(estudo);
    List<Tarefa> tarefas = estudo.getTarefas();

    Long tempo = 0L;
    Long cliques = 0L;
    Long percebida = 0L;
    Long medida = 0L;

    int numTarefas = 0;
    for (Tarefa tarefa : tarefas) {

        tempo += (tarefa.getFim().getTime() - tarefa.getInicio().getTime());

        cliques += tarefa.getCliques();

        if (tarefa.isConcluidoPercebido()) percebida++;
        if (tarefa.isConcluidoReal()) medida++;

        numTarefas++;
    }

    percebida = (percebida > 0) ? percebida * 100 / numTarefas : percebida;
    medida = (medida > 0) ? medida * 100 / numTarefas : medida;

    relatorio.setTempo(tempo/1000);
    relatorio.setCliques(cliques);
    relatorio.setPercebida(percebida);
    relatorio.setMedida(medida);

    Double satisfacao = estudo.getSus().getTotal();
    relatorio.setSatisfacao(satisfacao);

    if (estudo.getRelatorio() == null) {

        relatorio.setEstudo(estudo);
```

```
        estudo.setRelatorio(relatorio);
        relatorio.save();

    } else {

        relatorio.setEstudo(estudo);
        estudo.setRelatorio(relatorio);
        relatorio.update();
    }
}

@Authenticated(UsuarioAutenticado.class)
public Result iniciarTarefa() {

    Estudo estudo = estudoDAO.comToken(session(AUTH)).get();
    List<Evento> eventos = eventoDAO.mostraTodos();

    Tarefa tarefa = criarNovaTarefa(estudo);

    if(estudo.isTipo()) {

        return ok(estudo1portal.render(tarefa, tarefaForm, eventos));

    } else {

        Collections.shuffle(eventos);
        return ok(estudo0portal.render(tarefa, tarefaForm, eventos));
    }

}

@Authenticated(UsuarioAutenticado.class)
public Result concluirTarefa() {

    Tarefa form = tarefaForm.bindFromRequest().get();
    Tarefa tarefa = tarefaDAO.comId(form.getId()).get();

    if (tarefa.getFim() == null) {
```

```
        Calendar calendario = Calendar.getInstance();
        tarefa.setFim(calendario.getTime());
    }

    tarefa.setConcluidoPercebido(true);

    tarefa.update();

    Long idEstudo = tarefa.getEstudo().getId();
    Estudo estudo = estudoDAO.comId(idEstudo).get();

    return ok(relatorioTarefa.render(tarefa, tarefaForm, estudo.isTipo(),
        estudoForm));
}

@Authenticated(UsuarioAutenticado.class)
public Result desistirTarefa() {

    Tarefa form = tarefaForm.bindFromRequest().get();
    Tarefa tarefa = tarefaDAO.comId(form.getId()).get();

    if (tarefa.getFim() == null) {

        Calendar calendario = Calendar.getInstance();
        tarefa.setFim(calendario.getTime());
    }

    tarefa.setConcluidoPercebido(false);

    tarefa.update();

    Long idEstudo = tarefa.getEstudo().getId();
    Estudo estudo = estudoDAO.comId(idEstudo).get();

    return ok(relatorioTarefa.render(tarefa, tarefaForm, estudo.isTipo(),
        estudoForm));
}

@Authenticated(UsuarioAutenticado.class)
```



```
public Result enviarPesquisa() {

    Sus pesquisa = susForm.bindFromRequest().get();

    Estudo estudo = estudoDAO.comId(pesquisa.getId()).get();

    Long total = pesquisa.getQ1() - 1;
    total = total + 5 - pesquisa.getQ2();
    total = total + pesquisa.getQ3() - 1;
    total = total + 5 - pesquisa.getQ4();
    total = total + pesquisa.getQ5() - 1;
    total = total + 5 - pesquisa.getQ6();
    total = total + pesquisa.getQ7() - 1;
    total = total + 5 - pesquisa.getQ8();
    total = total + pesquisa.getQ9() - 1;
    total = total + 5 - pesquisa.getQ10();
    pesquisa.setTotal(total * 2.5);

    if (estudo.getSus() == null) {

        pesquisa.setEstudo(estudo);
        estudo.setSus(pesquisa);
        pesquisa.save();

    } else {

        pesquisa.setEstudo(estudo);
        estudo.setSus(pesquisa);
        pesquisa.update();
    }

    List<Tarefa> tarefas = estudo.getTarefas();
    Collections.sort(tarefas);

    gerarRelatorio(estudo);

    return ok(relatorioParcial.render(estudo.getTarefas(), estudo, estudoForm));
}
```

```
@Authenticated(UsuarioAutenticado.class)
public Result adicionaCliquesTarefa(Long id) {

    Tarefa tarefa = tarefaDAO.comId(id).get();

    if (tarefa.getFim() == null) {

        Long cliques = tarefa.getCliques();
        cliques++;
        tarefa.setCliques(cliques);

        tarefa.update();
    }

    return ok();
}

@Authenticated(UsuarioAutenticado.class)
public Result setConcluido(Long id) {

    Tarefa tarefa = tarefaDAO.comId(id).get();

    tarefa.setConcluidoReal(true);

    tarefa.update();

    return ok();
}

@Authenticated(UsuarioAutenticado.class)
public Tarefa criarNovaTarefa(Estudo estudo) {

    Tarefa tarefa;

    List<Tarefa> tarefas = estudo.getTarefas();

    String codigo = "EC" + (estudo.isTipo() ? "1" : "0");
    Calendar calendario = Calendar.getInstance();
```

```
    if (tarefas.size() == 0) {

        codigo = codigo + "1";

    } else {

        Tarefa ultimaTarefa = tarefas.get(tarefas.size() - 1);

        if (ultimaTarefa.getFim() == null) {

            return ultimaTarefa;

        } else {

            String numTarefa = Integer.toString(tarefas.size() + 1);
            codigo = codigo + numTarefa.toString();
        }

    }

    tarefa = new Tarefa(codigo, estudo, calendario.getTime());
    tarefa.save();

    return tarefa;
}

@Authenticated(UsuarioAutenticado.class)
public Estudo criarNovoEstudo(boolean tipo) {

    Calendar calendario;
    calendario = Calendar.getInstance();
    Usuario usuario = usuarioDAO.comToken(session(AUTH)).get();

    Estudo estudo = new Estudo(tipo, usuario, calendario.getTime());

    TokenSistema token = tokenSistemaDAO.comCodigo(session(AUTH)).get();
    estudo.setToken(token);

    return estudo;
}
```

```
    }  
}  
  
/*--- ControladorEventos.java ---*/  
  
package controllers;  
  
import java.util.Calendar;  
import java.util.Collections;  
import java.util.List;  
import java.util.TimeZone;  
  
import javax.inject.Inject;  
  
import autenticadores.UsuarioAutenticado;  
import daos.EstudoDAO;  
import daos.EventoDAO;  
import daos.TarefaDAO;  
import models.Estudo;  
import models.Evento;  
import models.Inscricao;  
import models.Tarefa;  
import play.data.Form;  
import play.data.FormFactory;  
import play.mvc.Controller;  
import play.mvc.Result;  
import play.mvc.Security.Authenticated;  
import views.html.*;  
  
public class ControladorEventos extends Controller {  
  
    @Inject  
    private EventoDAO eventoDAO;  
    @Inject  
    private TarefaDAO tarefaDAO;  
    @Inject  
    private EstudoDAO estudoDAO;  
  
    private Form<Tarefa> tarefaForm;
```

```
private Form<Inscricao> inscricaoForm;

public static final String AUTH = "auth";

@Inject
public ControladorEventos(FormFactory formFactory) {

    this.tarefaForm = formFactory.form(Tarefa.class);
    this.inscricaoForm = formFactory.form(Inscricao.class);
}

public Result detalhar() {

    Tarefa form = tarefaForm.bindFromRequest().get();
    Tarefa tarefa = tarefaDAO.comId(form.getId()).get();

    Long cliquesForm = form.getCliques();
    if (cliquesForm > tarefa.getCliques()) tarefa.setCliques(cliquesForm);
    tarefa.update();

    Evento evento = eventoDAO.comId(form.getEvento()).get();
    evento.setPrograma(calcularPrograma(evento));

    Estudo estudo = tarefa.getEstudo();
    if(estudo.isTipo()) {

        int[] datas = calcularDataEvento(evento);
        String periodo = dataEmString(datas, "-");

        return ok(estudo1evento.render(tarefa, tarefaForm, evento, inscricaoForm,
            periodo));
    }

    return ok(estudo0evento.render(tarefa, tarefaForm, evento));
}

private String calcularPrograma(Evento evento) {

    int[] datas = calcularDataEvento(evento);
```

```
String dia1 = Integer.toString(datas[0]) + "/" + Integer.toString(datas[1]);
String dia2 = Integer.toString(datas[0] + 1) + "/" +
    Integer.toString(datas[1]);
String dia3 = Integer.toString(datas[0] + 2) + "/" +
    Integer.toString(datas[1]);

String programa = evento.getPrograma();

programa = programa.replace("@dia1", dia1);
programa = programa.replace("@dia2", dia2);
programa = programa.replace("@dia3", dia3);

return programa;
}

public Result mostrarPrograma() {

    Tarefa form = tarefaForm.bindFromRequest().get();
    Tarefa tarefa = tarefaDAO.comId(form.getId()).get();
    Evento evento = eventoDAO.comId(form.getEvento()).get();

    String programa = calcularPrograma(evento);

    Estudo estudo = tarefa.getEstudo();
    if(estudo.isTipo()) {

        return ok(estudo1programa.render(tarefa, tarefaForm, evento.getNome(),
            programa));
    }

    return ok(estudo0programa.render(tarefa,tarefaForm, evento.getSigla(),
        programa));
}

public Result mostrarLocal() {

    Tarefa form = tarefaForm.bindFromRequest().get();
    Tarefa tarefa = tarefaDAO.comId(form.getId()).get();
```

```
Evento evento = eventoDAO.comId(form.getEvento()).get();

int[] datas = calcularDataEvento(evento);

String dataInicial = Integer.toString(datas[0]);
String dataFinal = Integer.toString(datas[0] + 2) +
    "/" + Integer.toString(datas[1]) +
    "/" + Integer.toString(datas[2]);

return ok(estudo0local.render(tarefa, tarefaForm, evento, dataInicial,
    dataFinal));
}

public Result mostrarInformacoes() {

    Tarefa form = tarefaForm.bindFromRequest().get();
    Tarefa tarefa = tarefaDAO.comId(form.getId()).get();
    Evento evento = eventoDAO.comId(form.getEvento()).get();
    Estudo estudo = tarefa.getEstudo();

    if(estudo.isTipo()) {

        return ok(estudo1informacoes.render(tarefa, tarefaForm, evento));
    }

    return ok(estudo0informacoes.render(tarefa, tarefaForm, evento));
}

public Result mostrarPagamento() {

    Tarefa form = tarefaForm.bindFromRequest().get();
    Tarefa tarefa = tarefaDAO.comId(form.getId()).get();
    Evento evento = eventoDAO.comId(form.getEvento()).get();

    String data = calcularDataPagamento(evento);

    return ok(estudo0pagamento.render(tarefa, tarefaForm, evento, data));
}
```

```
public Result mostrarParticipe() {

    Tarefa form = tarefaForm.bindFromRequest().get();
    Tarefa tarefa = tarefaDAO.comId(form.getId()).get();
    Evento evento = eventoDAO.comId(form.getEvento()).get();

    Estudo estudo = tarefa.getEstudo();
    if(estudo.isTipo()) {

        String data = calcularDataPagamento(evento);
        return ok(estudo1participe.render(tarefa, tarefaForm, inscricaoForm,
            evento, data));
    }

    return ok(estudo0participe.render(tarefa, tarefaForm, inscricaoForm,
        evento));
}

public Result mostrarInscricao() {

    Inscricao form = inscricaoForm.bindFromRequest().get();
    Tarefa tarefa = tarefaDAO.comId(form.getTarefa()).get();
    Evento evento = eventoDAO.comId(form.getEvento()).get();

    return ok(estudo1inscricao.render(tarefa, tarefaForm, inscricaoForm,
        evento));
}

public Result mostrarCertificadoForm() {

    Tarefa form = tarefaForm.bindFromRequest().get();
    Tarefa tarefa = tarefaDAO.comId(form.getId()).get();
    Evento evento = eventoDAO.comId(form.getEvento()).get();

    Estudo estudo = tarefa.getEstudo();
    if(estudo.isTipo()) {

        String data = calcularDataPagamento(evento);
        return ok(estudo1participe.render(tarefa, tarefaForm, inscricaoForm,
```



```
        evento, data));
    }

    return ok(estudo0certificadoForm.render(tarefa, tarefaForm, inscricaoForm,
        evento));
}

public Result mostrarCertificado() {

    Inscricao form = inscricaoForm.bindFromRequest().get();

    Tarefa tarefa = tarefaDAO.comId(form.getTarefa()).get();
    Evento evento = eventoDAO.comId(form.getEvento()).get();

    boolean participou = false;

    if (evento.getSigla().equals("SBSI")) {

        if (testarCertificado(form)) {

            tarefa.setConcluidoReal(true);
            tarefa.update();
            participou = true;
        }
    }

    Estudo estudo = estudoDAO.comToken(session(AUTH)).get();

    if (estudo.isTipo()) {

        return ok(estudo1certificado.render(tarefa, tarefaForm, evento,
            participou));
    }

    return ok(estudo0certificado.render(tarefa, tarefaForm, evento,
        participou));
}

public int[] calcularDataEvento(Evento evento) {
```

```
Calendar hoje = Calendar.getInstance(TimeZone.getDefault());

int data = evento.getData();
int passado = hoje.get(Calendar.DAY_OF_YEAR) + data;

passado = (passado < 0) ? (passado / 365) - 1 : 0;

int dia = Math.abs(hoje.get(Calendar.DAY_OF_YEAR) + data) % 30;
int mes = Math.abs((hoje.get(Calendar.DAY_OF_YEAR) + data) / 30) + 1;
int ano = hoje.get(Calendar.YEAR) + passado;

if (dia == 0) dia++;
if (dia > 28) dia = dia - 2;

int[] datas = {dia, mes, ano};

return datas;
}

private String dataEmString(int[] datas, String separador) {

return Integer.toString(datas[0]) +
    " " + separador + " " +
    Integer.toString(datas[0] + 2) + "/" +
    Integer.toString(datas[1]) + "/" +
    Integer.toString(datas[2]);
}

public String calcularDataPagamento(Evento evento) {

int[] datas = calcularDataEvento(evento);

return Integer.toString(datas[0]) +
    "/" + Integer.toString(datas[1] - 2) +
    "/" + Integer.toString(datas[2]);
}

@Authenticated(UsuarioAutenticado.class)
```

```
public Result fazerInscricao() {

    Inscricao form = inscricaoForm.bindFromRequest().get();

    Tarefa tarefa = tarefaDAO.comId(form.getTarefa()).get();
    Evento evento = eventoDAO.comId(form.getEvento()).get();
    Estudo estudo = tarefa.getEstudo();
    List<Evento> eventos = eventoDAO.mostraTodos();

    if (testarInscricao(form)) {

        tarefa.setConcluidoReal(true);
        tarefa.update();
    }

    if(estudo.isTipo()) {

        return ok(estudo1portal.render(tarefa, tarefaForm, eventos));

    } else {

        Collections.shuffle(eventos);
        return ok(estudo0portal.render(tarefa, tarefaForm, eventos));
    }
}

@Authenticated(UsuarioAutenticado.class)
private boolean testarInscricao(Inscricao form) {

    if (!form.getNumCartao().replaceAll("\\s+", "").equals("4609868766944752"))
        {return false;}
    else if (!form.getValidade().equals("09/2030")) {return false;}
    else if (!form.getCodigoSeguranca().equals("902")) {return false;}

    return true;
}

private boolean testarCertificado(Inscricao form) {
```

```
        if (!form.getNome().toLowerCase().equals("martin fowler")) {return false;}
        else if (!form.getEmail().toLowerCase().equals("martin@martinfowler.com"))
            {return false;}
        else if (!form.getCpf().equals("683.563.770-64")) {return false;}
        else if (!form.getEndereco().equals("1234")) {return false;}

        return true;
    }
}
```

```
/*--- UsuarioAutenticado.java ---*/
```

```
package autenticadores;
```

```
import controllers.*;
```

```
import daos.*;
```

```
import models.*;
```

```
import java.util.Optional;
```

```
import javax.inject.Inject;
```

```
import play.mvc.Http.Context;
```

```
import play.mvc.*;
```

```
import play.mvc.Security.Authenticator;
```

```
public class UsuarioAutenticado extends Authenticator {
```

```
    @Inject
```

```
    private UsuarioDAO usuarioDAO;
```

```
    @Override
```

```
    public String getUsername(Context context) {
```

```
        String codigo = context.session().get(ControladorUsuario.AUTH);
```

```
        Optional<Usuario> possivelUsuario = usuarioDAO.comToken(codigo);
```

```
        if(possivelUsuario.isPresent()) {
```

```
            return possivelUsuario.get().getEmail();
```

```
    }

    return null;
}

@Override
public Result onUnauthorized(Context context) {

    context.flash().put("danger", "Nao autorizado!");

    return redirect(routes.HomeController.index());
}
}
```

```
/*--- Derivador.java ---*/
```

```
package validadores;
```

```
import org.apache.commons.codec.binary.Hex;
```

```
import javax.crypto.SecretKeyFactory;
```

```
import javax.crypto.spec.PBEKeySpec;
```

```
import java.security.NoSuchAlgorithmException;
```

```
import java.security.SecureRandom;
```

```
import java.util.Scanner;
```

```
import javax.crypto.SecretKey;
```

```
public class Derivador {
```

```
    /**
```

```
     * Gerar chave derivada da senha
```

```
     * @param key
```

```
     * @param salt
```

```
     * @param iterations
```

```
     * @return
```

```
     */
```

```
    public static String generateDerivedKey(
```

```
        String password, String salt, Integer iterations) {
```

```
        PBEKeySpec spec = new PBEKeySpec(password.toCharArray(), salt.getBytes(),
```

```
        iterations, 128);
    SecretKeyFactory pbkdf2 = null;
    String derivedPass = null;
    try {
        pbkdf2 = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA512");
        SecretKey sk = pbkdf2.generateSecret(spec);
        derivedPass = Hex.encodeHexString(sk.getEncoded());
    } catch (Exception e) {
        e.printStackTrace();
    }
    return derivedPass;
}

/*Usado para gerar o salt */
public String getSalt() throws NoSuchAlgorithmException {

    SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
    //SecureRandom sr = new SecureRandom();
    byte[] salt = new byte[16];
    sr.nextBytes(salt);
    return Hex.encodeHexString(salt);
}

public String deriva(String str) throws NoSuchAlgorithmException {

    Derivador obj = new Derivador();

    String salt;
    int it = 10000;

    salt = obj.getSalt();

    System.out.println("Senha original = " + str);
    System.out.println("Sal gerado = " + salt);
    System.out.println("Numero de iteracoes = " + it);

    String chaveDerivada = generateDerivedKey(str, salt, it);

    System.out.println("Chave derivada da senha = " + chaveDerivada );
```

```
        return chaveDerivada;
    }
}



---


/*--- Encriptador.java ---*/

package validadores;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class Encriptador {

    private static MessageDigest digester;

    static {
        try {
            digester = MessageDigest.getInstance("MD5");
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
    }

    public static String crypt(String str) {
        if (str == null || str.length() == 0) {
            throw new IllegalArgumentException("String to encrypt cannot be null or
                zero length");
        }

        digester.update(str.getBytes());
        byte[] hash = digester.digest();
        StringBuffer hexString = new StringBuffer();
        for (int i = 0; i < hash.length; i++) {
            if ((0xff & hash[i]) < 0x10) {
                hexString.append("0" + Integer.toHexString((0xFF & hash[i])));
            } else {

```

```
        hexString.append(Integer.toHexString(0xFF & hash[i]));
    }
}
return hexString.toString();
}
}
```

```
/*--- Utils.java ---*/
```

```
package validadores;
```

```
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.SecureRandom;
```

```
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;
```

```
/**
 * General utilities for the third chapter examples.
 */
```

```
public class Utils
```

```
{
```

```
    private static String digits = "0123456789abcdef";
```

```
/**
 * Return length many bytes of the passed in byte array as a hex string.
 *
 * @param data the bytes to be converted.
 * @param length the number of bytes in the data block to be converted.
 * @return a hex representation of length bytes of data.
 */
```

```
public static String toHex(byte[] data, int length)
```

```
{
```

```
    StringBuffer buf = new StringBuffer();
```



```
    for (int i = 0; i != length; i++)
    {
        int v = data[i] & 0xff;

        buf.append(digits.charAt(v >> 4));
        buf.append(digits.charAt(v & 0xf));
    }

    return buf.toString();
}

/**
 * Return the passed in byte array as a hex string.
 *
 * @param data the bytes to be converted.
 * @return a hex representation of data.
 */
public static String toHex(byte[] data)
{
    return toHex(data, data.length);
}

/**
 * Create a key for use with AES.
 *
 * @param bitLength
 * @param random
 * @return an AES key.
 * @throws NoSuchAlgorithmException
 * @throws NoSuchProviderException
 */
public static SecretKey createKeyForAES(
    int bitLength,
    SecureRandom random)
    throws NoSuchAlgorithmException, NoSuchProviderException
{
    KeyGenerator generator = KeyGenerator.getInstance("AES");
```

```
        generator.init(128, random);

        return generator.generateKey();
    }

    /**
     * Create an IV suitable for using with AES in CTR mode.
     * <p>
     * The IV will be composed of 4 bytes of message number,
     * 4 bytes of random data, and a counter of 8 bytes.
     *
     * @param messageNumber the number of the message.
     * @param random a source of randomness
     * @return an initialised IvParameterSpec
     */
    public static IvParameterSpec createCtrIvForAES(
        int            messageNumber,
        SecureRandom  random)
    {
        byte[]        ivBytes = new byte[16];

        // initially randomize

        random.nextBytes(ivBytes);

        // set the message number bytes

        ivBytes[0] = (byte)(messageNumber >> 24);
        ivBytes[1] = (byte)(messageNumber >> 16);
        ivBytes[2] = (byte)(messageNumber >> 8);
        ivBytes[3] = (byte)(messageNumber >> 0);

        // set the counter bytes to 1

        for (int i = 0; i != 7; i++)
        {
            ivBytes[8 + i] = 0;
        }
    }
}
```

```
        ivBytes[15] = 1;

        return new IvParameterSpec(ivBytes);
    }

    /**
     * Convert a byte array of 8 bit characters into a String.
     *
     * @param bytes the array containing the characters
     * @param length the number of bytes to process
     * @return a String representation of bytes
     */
    public static String toString(
        byte[] bytes,
        int length)
    {
        char[] chars = new char[length];

        for (int i = 0; i != chars.length; i++)
        {
            chars[i] = (char)(bytes[i] & 0xff);
        }

        return new String(chars);
    }

    /**
     * Convert a byte array of 8 bit characters into a String.
     *
     * @param bytes the array containing the characters
     * @return a String representation of bytes
     */
    public static String toString(
        byte[] bytes)
    {
        return toString(bytes, bytes.length);
    }

    /**
```

```
* Convert the passed in String to a byte array by
* taking the bottom 8 bits of each character it contains.
*
* @param string the string to be converted
* @return a byte array representation
*/
public static byte[] toByteArray(
    String string)
{
    byte[] bytes = new byte[string.length()];
    char[] chars = string.toCharArray();

    for (int i = 0; i != chars.length; i++)
    {
        bytes[i] = (byte)chars[i];
    }

    return bytes;
}
}
```

```
/*---- Validador.java ----*/
```

```
package validadores;

import models.*;

import java.util.Optional;

import javax.inject.Inject;

import daos.UsuarioDAO;
import play.data.Form;

public class Validador {

    private UsuarioDAO usuarioDAO;

    @Inject
```

```
public Validador(UsuarioDAO usuarioDAO) {

    this.usuarioDAO = usuarioDAO;
}

public boolean temErros(Form<Usuario> form) {

    validaSenha(form);
    return form.hasErrors();
}

private void validaSenha(Form<Usuario> form) {
    Optional<String> senha = form.field("senha").getValue();
}

}



---




---


/*--- EstudoDAO.java ---*/

package daos;

import models.*;

import java.util.*;
import io.ebean.Finder;

public class EstudoDAO {

    private static Finder<Long, Estudo> find = new Finder<>(Estudo.class);

    public List<Estudo> mostraTodos() {

        return find.all();
    }

    public Optional<Estudo> comId(Long id) {

        Estudo estudo = find
            .query()
```

```
        .where()
        .eq("id", id)
        .findOne();

    return Optional.ofNullable(estudo);
}

public Optional<Estudo> comToken(String codigo) {

    Estudo estudo = find
        .query()
        .where()
        .eq("token.codigo", codigo)
        .findOne();

    return Optional.ofNullable(estudo);
}
}



---




---


/*--- EventoDAO.java ---*/

package daos;

import models.*;

import java.util.*;
import io.ebean.Finder;

public class EventoDAO {

    private static Finder<Long, Evento> find = new Finder<>(Evento.class);

    public List<Evento> mostraTodos() {

        List<Evento> eventos = find.all();

        Collections.sort(eventos);
    }
}
```

```
        return eventos;
    }

    public Optional<Evento> comId(Long id) {

        Evento evento = find
            .query()
            .where()
            .eq("id", id)
            .findOne();

        return Optional.ofNullable(evento);
    }
}



---




---


/*---- TarefaDAO.java ----*/

package daos;

import models.*;

import java.util.*;
import io.ebean.Finder;

public class TarefaDAO {

    private static Finder<Long, Tarefa> find = new Finder<>(Tarefa.class);

    public List<Tarefa> mostraTodos() {

        return find.all();
    }

    public Optional<Tarefa> comId(Long id) {

        Tarefa tarefa = find
            .query()
            .where()
```

```
        .eq("id", id)
        .findOne();

    return Optional.ofNullable(tarefa);
}

}



---


/*--- TokenCadastroDAO.java ---*/

package daos;

import models.*;

import java.util.Optional;
import io.ebean.Finder;

public class TokenCadastroDAO {

    private static Finder<Long, TokenCadastro> tokens = new
        Finder<>(TokenCadastro.class);

    public static Optional<TokenCadastro> comCodigo(String codigo) {

        TokenCadastro token = tokens
            .query()
            .where()
            .eq("codigo", codigo)
            .findOne();
        return Optional.ofNullable(token);
    }

}



---


/*--- TokenSistemaDAO.java ---*/

package daos;

import java.util.Optional;
```



```
import io.ebean.Finder;
import models.TokenSistema;

public class TokenSistemaDAO {

    private static Finder<Long, TokenSistema> find = new
        Finder<>(TokenSistema.class);

    public Optional<TokenSistema> comCodigo(String codigo) {

        TokenSistema token = find
            .query()
            .where()
            .eq("codigo", codigo)
            .findOne();

        return Optional.ofNullable(token);
    }

    public Optional<TokenSistema> comId(Long id) {

        TokenSistema token = find.byId(id);

        return Optional.ofNullable(token);

    }

}
```

```
/*--- UsuarioDAO.java ---*/
```

```
package daos;

import models.*;

import java.util.*;
import io.ebean.Finder;
```

```
public class UsuarioDAO {

    private static Finder<Long, Usuario> find = new Finder<>(Usuario.class);

    public Optional<Usuario> comEmail(String email) {

        Usuario usuario = find
            .query()
            .where()
            .eq("email", email)
            .findOne();

        return Optional.ofNullable(usuario);
    }

    public Optional<Usuario> comEmailSenha(String email, String criptoSenha) {

        Usuario usuario = find
            .query()
            .where()
            .eq("email", email)
            .eq("senha", criptoSenha)
            .findOne();

        return Optional.ofNullable(usuario);
    }

    public List<Usuario> mostraTodos() {

        return find.all();
    }

    public Optional<Usuario> comToken(String codigo) {

        Usuario usuario = find
            .query()
            .where()
            .eq("token.codigo", codigo)
            .findOne();
    }
}
```

```
        return Optional.ofNullable(usuario);
    }
}



---




---


/*--- EmailCadastro.java ---*/

package models;

import play.libs.mailer.Email;

public class EmailCadastro extends Email {

    private static final String REMETENTE = "UsabilidadeZero <no-reply@ufsc.br>";
    private static final String ASSUNTO = "Confirmacao de cadastro na API de
        produtos!";
    private static final String CORPO_FORMAT = "Ola, %s! Por favor clique no link
        a seguir para confirmar seu cadastro! <a href='%s'>Confirmar
        cadastro!</a>";

    public EmailCadastro(TokenCadastro token) {

        Usuario usuario = token.getUsuario();
        String destinatario = String.format("<%s>", usuario.getEmail());
        String link =
            String.format("http://localhost:9000/usuario/confirma/%s/%s",
                usuario.getEmail(), token.getCodigo());
        String corpo = String.format(CORPO_FORMAT, usuario.getEmail(), link);

        this.addTo(destinatario);
        this.setFrom(REMETENTE);
        this.setSubject(ASSUNTO);
        this.setBodyHtml(corpo);
    }
}



---




---


/*--- Estudo.java ---*/
```

```
package models;

import java.util.Collections;
import java.util.Date;
import java.util.List;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.OneToOne;

import io.ebean.Finder;
import io.ebean.Model;
import play.data.validation.Constraints;

@Entity
public class Estudo extends Model {

    private static final Long serialVersionUID = 1L;

    @Id
    public Long id;
    @Constraints.Required
    public boolean tipo;
    @ManyToOne
    public Usuario usuario;
    public Date inicio;
    public Date fim;
    @OneToMany(mappedBy = "estudo")
    public List<Tarefa> tarefas;
    @OneToOne(mappedBy = "estudo")
    public Sus sus;
    @OneToOne(mappedBy = "estudo")
    public RelatorioEstudo relatorio;
    @OneToOne
    private TokenSistema token;
    @OneToOne
```

```
private Estudo relacionado;

public static Finder<Long, Estudo> find = new Finder<>(Estudo.class);

public Estudo(boolean tipo, Usuario usuario, Date inicio) {

    this.tipo = tipo;
    this.usuario = usuario;
    this.inicio = inicio;
}

public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}

public boolean isTipo() {
    return tipo;
}

public void setTipo(boolean tipo) {
    this.tipo = tipo;
}

public Usuario getUsuario() {
    return usuario;
}

public Date getInicio() {
    return inicio;
}

public Date getFim() {
    return fim;
}
```

```
public void setFim(Date fim) {
    this.fim = fim;
}

public List<Tarefa> getTarefas() {

    Collections.sort(tarefas);
    return tarefas;
}

public void setTarefas(List<Tarefa> tarefas) {
    this.tarefas = tarefas;
}

public Sus getSus() {
    return sus;
}

public void setSus(Sus sus) {
    this.sus = sus;
}

public TokenSistema getToken() {
    return token;
}

public void setToken(TokenSistema token) {
    this.token = token;
}

public RelatorioEstudo getRelatorio() {
    return relatorio;
}

public void setRelatorio(RelatorioEstudo relatorio) {
    this.relatorio = relatorio;
}

public Estudo getRelacionado() {
```

```
        return relacionado;
    }

    public void setRelacionado(Estudo relacionado) {
        this.relacionado = relacionado;
    }
}



---




---


/*--- Evento.java ---*/

package models;

import java.util.*;
import javax.persistence.*;

import io.ebean.*;
import play.data.validation.Constraints.*;

@Entity
public class Evento extends Model implements Comparable<Evento> {

    private static final Long serialVersionUID = 1L;

    @Id @GeneratedValue
    private Long id;
    private String nome;
    private String sigla;
    @Column(columnDefinition = "TEXT")
    private String descricao;
    @Column(columnDefinition = "TEXT")
    private String programa;
    @Column(columnDefinition = "TEXT")
    private String informacoes;
    private String local;
    private int data;
    private double preco;
    private String categoria;

    public Long getId() {
```

```
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getSigla() {
        return sigla;
    }
    public void setSigla(String sigla) {
        this.sigla = sigla;
    }
    public String getDescricao() {
        return descricao;
    }
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
    public String getPrograma() {
        return programa;
    }
    public void setPrograma(String programa) {
        this.programa = programa;
    }
    public String getInformacoes() {
        return informacoes;
    }
    public void setInformacoes(String informacoes) {
        this.informacoes = informacoes;
    }
    public String getLocal() {
        return local;
    }
    public void setLocal(String local) {
```



```
        this.local = local;
    }
    public int getData() {
        return data;
    }
    public void setData(int data) {
        this.data = data;
    }
    public double getPreco() {
        return preco;
    }
    public void setPreco(double preco) {
        this.preco = preco;
    }
    public String getCategoria() {
        return categoria;
    }
    public void setCategoria(String categoria) {
        this.categoria = categoria;
    }
    @Override
    public int compareTo(Evento e) {

        return this.getData() - e.getData();
    }
}
```

```
/*--- Inscricao.java ---*/
```

```
package models;
```

```
import play.data.validation.Constraints;
```

```
import play.data.validation.Constraints.Validate;
```

```
public class Inscricao {
```

```
    private Long tarefa;
```

```
    private Long evento;
```

```
    private String nome;
```

```
private String email;
private String fone;
private String cpf;
private String endereco;
private String cidade;
private String numCartao;
private String titularCartao;
private String validade;
private String codigoSeguranca;
private boolean valor;

public Long getTarefa() {
    return tarefa;
}

public void setTarefa(Long tarefa) {
    this.tarefa = tarefa;
}

public Long getEvento() {
    return evento;
}

public void setEvento(Long evento) {
    this.evento = evento;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getEmail() {
    return email;
}
```

```
public void setEmail(String email) {
    this.email = email;
}

public String getFone() {
    return fone;
}

public void setFone(String fone) {
    this.fone = fone;
}

public String getCpf() {
    return cpf;
}

public void setCpf(String cpf) {
    this.cpf = cpf;
}

public String getEndereco() {
    return endereco;
}

public void setEndereco(String endereco) {
    this.endereco = endereco;
}

public String getCidade() {
    return cidade;
}

public void setCidade(String cidade) {
    this.cidade = cidade;
}

public String getNumCartao() {
    return numCartao;
}
```

```
public void setNumCartao(String numCartao) {
    this.numCartao = numCartao;
}

public String getTitularCartao() {
    return titularCartao;
}

public void setTitularCartao(String titularCartao) {
    this.titularCartao = titularCartao;
}

public String getValidade() {
    return validade;
}

public void setValidade(String validade) {
    this.validade = validade;
}

public String getCodigoSeguranca() {
    return codigoSeguranca;
}

public void setCodigoSeguranca(String codigoSeguranca) {
    this.codigoSeguranca = codigoSeguranca;
}

public boolean isValor() {
    return valor;
}

public void setValor(boolean valor) {
    this.valor = valor;
}
}
```

```
/*--- RelatorioEstudo.java ---*/
```

```
package models;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToOne;

import io.ebean.Model;

@Entity
public class RelatorioEstudo extends Model {

    private static final Long serialVersionUID = 1L;

    @Id
    public Long id;
    @OneToOne
    public Estudo estudo;
    public Long tempo;
    public Long cliques;
    public Long percebida;
    public Long medida;
    public Double satisfacao;

    public RelatorioEstudo(Estudo estudo) {

        this.estudo = estudo;
    }

    public Estudo getEstudo() {
        return estudo;
    }

    public void setEstudo(Estudo estudo) {
        this.estudo = estudo;
    }

    public Long getTempo() {
        return tempo;
    }
}
```

```
}

public void setTempo(Long tempo) {
    this.tempo = tempo;
}

public Long getCliques() {
    return cliques;
}

public void setCliques(Long cliques) {
    this.cliques = cliques;
}

public Long getPercebida() {
    return percebida;
}

public void setPercebida(Long percebida) {
    this.percebida = percebida;
}

public Long getMedida() {
    return medida;
}

public void setMedida(Long medida) {
    this.medida = medida;
}

public Double getSatisfacao() {
    return satisfacao;
}

public void setSatisfacao(Double satisfacao) {
    this.satisfacao = satisfacao;
}
}
```

```
/*--- Sus.java ---*/

package models;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToOne;

import io.ebean.Model;

@Entity
public class Sus extends Model {

    private static final Long serialVersionUID = 1L;

    @Id
    public Long id;
    @OneToOne
    public Estudo estudo;
    public Long q1;
    public Long q2;
    public Long q3;
    public Long q4;
    public Long q5;
    public Long q6;
    public Long q7;
    public Long q8;
    public Long q9;
    public Long q10;
    public Double total;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }
}
```

```
public Estudo getEstudo() {
    return estudo;
}

public void setEstudo(Estudo estudo) {
    this.estudo = estudo;
}

public Long getQ1() {
    return q1;
}

public void setQ1(Long q1) {
    this.q1 = q1;
}

public Long getQ2() {
    return q2;
}

public void setQ2(Long q2) {
    this.q2 = q2;
}

public Long getQ3() {
    return q3;
}

public void setQ3(Long q3) {
    this.q3 = q3;
}

public Long getQ4() {
    return q4;
}

public void setQ4(Long q4) {
    this.q4 = q4;
}
```



```
public Long getQ5() {
    return q5;
}

public void setQ5(Long q5) {
    this.q5 = q5;
}

public Long getQ6() {
    return q6;
}

public void setQ6(Long q6) {
    this.q6 = q6;
}

public Long getQ7() {
    return q7;
}

public void setQ7(Long q7) {
    this.q7 = q7;
}

public Long getQ8() {
    return q8;
}

public void setQ8(Long q8) {
    this.q8 = q8;
}

public Long getQ9() {
    return q9;
}

public void setQ9(Long q9) {
    this.q9 = q9;
}
```

```
    }

    public Long getQ10() {
        return q10;
    }

    public void setQ10(Long q10) {
        this.q10 = q10;
    }

    public Double getTotal() {
        return total;
    }

    public void setTotal(Double total) {
        this.total = total;
    }
}



---




---


/*--- Tarefa.java ---*/

package models;

import java.util.Date;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToOne;

import io.ebean.Finder;
import io.ebean.Model;

@Entity
public class Tarefa extends Model implements Comparable<Tarefa> {

    private static final Long serialVersionUID = 1L;

    @Id
    public Long id;
```

```
public String codigo;
@ManyToOne
public Estudo estudo;
public Long evento;
public Date inicio;
public Date fim;
public Long cliques;
public boolean concluidoReal;
public boolean concluidoPercebido;

public static Finder<Long, Tarefa> find = new Finder<>(Tarefa.class);

public Tarefa(String codigo, Estudo estudo, Date inicio) {

    this.codigo = codigo;
    this.estudo = estudo;
    this.inicio = inicio;
    this.cliques = (long) 0;
    this.concluidoReal = false;
    this.concluidoPercebido = false;
}

public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}

public String getCodigo() {
    return codigo;
}

public void setCodigo(String codigo) {
    this.codigo = codigo;
}

public Estudo getEstudo() {
```

```
        return estudo;
    }

    public void setEstudo(Estudo estudo) {
        this.estudo = estudo;
    }

    public Date getInicio() {
        return inicio;
    }

    public void setInicio(Date inicio) {
        this.inicio = inicio;
    }

    public Date getFim() {
        return fim;
    }

    public void setFim(Date fim) {
        this.fim = fim;
    }

    public Long getCliques() {
        return cliques;
    }

    public void setCliques(Long cliques) {
        this.cliques = cliques;
    }

    public boolean isConcluidoReal() {
        return concluidoReal;
    }

    public void setConcluidoReal(boolean concluidoReal) {
        this.concluidoReal = concluidoReal;
    }
}
```

```
public boolean isConcluidoPercebido() {
    return concluidoPercebido;
}

public void setConcluidoPercebido(boolean concluidoPercebido) {
    this.concluidoPercebido = concluidoPercebido;
}

public Long getEvento() {
    return evento;
}

public void setEvento(Long evento) {
    this.evento = evento;
}

@Override
public int compareTo(Tarefa t) {

    return (int) (this.getId() - t.getId());
}
}
```

```
/*--- TokenCadastro.java ---*/
```

```
package models;

import validadores.*;

import javax.persistence.*;
import io.ebean.Model;
import java.security.SecureRandom;

@Entity
public class TokenCadastro extends Model {

    SecureRandom random;

    @Id @GeneratedValue
```

```
private Long id;
@OneToOne
private Usuario usuario;
private String codigo;

public TokenCadastro(Usuario usuario) {

    this.usuario = usuario;
    this.codigo =
        Encriptador.crypt(usuario.getId()+usuario.getEmail()+this.gerarToken());
}

public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}

public Usuario getUsuario() {
    return usuario;
}

public void setUsuario(Usuario usuario) {
    this.usuario = usuario;
}

public String getCodigo() {
    return codigo;
}

public void setCodigo(String codigo) {
    this.codigo = codigo;
}

public String gerarToken() {
    this.random = new SecureRandom();
    byte bytes[] = new byte[20];
```

```
        random.nextBytes(bytes);
        return bytes.toString();
    }
}



---




---


/*--- TokenSistema.java ---*/

package models;

import java.security.SecureRandom;
import java.util.Date;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.OneToOne;

import io.ebean.Model;
import validadores.Encriptador;

@Entity
public class TokenSistema extends Model {

    private SecureRandom random;
    @Id @GeneratedValue
    private long id;
    @OneToOne
    private Usuario usuario;
    @OneToOne(mappedBy = "token")
    private Estudo estudo;
    private String codigo;
    private Date expiracao;

    public TokenSistema(Usuario usuario) {

        this.usuario = usuario;
        this.expiracao = new Date();
    }
}
```

```
        this.codigo =
            Encriptador.crypt(expiracao.toString()+usuario.toString()+this.gerarToken());
    }

    public long getId() {
        return id;
    }

    public Usuario getUsuario() {
        return usuario;
    }

    public Estudo getEstudo() {
        return estudo;
    }

    public void setEstudo(Estudo estudo) {
        this.estudo = estudo;
    }

    public String getCodigo() {
        return codigo;
    }

    public Date getExpiracao() {
        return expiracao;
    }

    public String gerarToken() {
        this.random = new SecureRandom();
        byte bytes[] = new byte[20];
        random.nextBytes(bytes);
        return bytes.toString();
    }
}



---




---


/*--- Usuario.java ---*/
```



```
package models;

import java.util.List;

import javax.persistence.*;

import io.ebean.*;
import play.data.validation.Constraints.*;

@Entity
public class Usuario extends Model {

    private static final Long serialVersionUID = 1L;

    @Id @GeneratedValue
    private Long id;
    @Required(message = "Voce precisa fornecer um email valido!")
    private String email;
    @Required(message = "Voce precisa fornecer uma senha!")
    private String senha;
    private boolean verificado;
    @OneToOne(mappedBy = "usuario")
    private TokenSistema token;
    @OneToMany(mappedBy = "usuario")
    private List<Estudo> estudos;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
```

```
        this.email = email;
    }

    public String getSenha() {
        return senha;
    }

    public void setSenha(String senha) {
        this.senha = senha;
    }

    public boolean isVerificado() {
        return verificado;
    }

    public void setVerificado(boolean verificado) {
        this.verificado = verificado;
    }

    public TokenSistema getToken() {
        return token;
    }

    public void setToken(TokenSistema token) {
        this.token = token;
    }
}
```

```
/*--- concluir-tarefa.js ---*/
```

```
$("#concluir-tarefa").attr("disabled", true);
$("#desistir-tarefa").attr("disabled", true);
```

```
/*--- contador-cliques.js ---*/
```

```
var idTarefa = $("#id-tarefa").text();
var cliquesTarefa = 0;
```

```
//Adiciona cliques
```

```
function adicionaClique() {

    cliquesTarefa++;

    $('input[name="cliques"]').val(cliquesTarefa);
    $("#contador").text(cliquesTarefa);

    $.get(jsRoutes.controllers.ControladorEstudos.adicionaCliquesTarefa(idTarefa));
}

$(document).click(function(e) {

    adicionaClique();
});

$('body').backDetect(function(){

    adicionaClique();
});



---




---


/*--- embaralha-botoes.js ---*/

$(function () {
    var parent = $(".embaralha");
    var forms = parent.children();
    while (forms.length) {
        parent.append(forms.splice(Math.floor(Math.random() * forms.length),
            1)[0]);
    }
});



---




---


/*--- filtro.js ---*/

$(document).ready(function(){

    // Iniciando isotope
    var gridProximos = $('.grid-proximos');
    var gridPassados = $('.grid-passados');
```

```
iniciaAnimacao(gridProximos);
iniciaAnimacao(gridPassados);

function iniciaAnimacao(grid) {

    grid.isotope({
        itemSelector: '#myDIV',
        layoutMode: 'fitRows',
        getSortData: {
            sigla: '.sigla',
            nome: '.nome',
            local: '.local',
            dataEvento: '.data-evento'
        }
    });
}

// Filtrando eventos
$('#input-filtro').on("keyup", function() {

    filtrar(this, gridProximos);
    filtrar(this, gridPassados);
})

function filtrar(input, grid) {

    var value = $(input).val().toLowerCase();

    grid.isotope({

        filter: function() {

            return $(this).text().toLowerCase().indexOf(value) > -1;
        }
    })
}

// Ordenando eventos
$('.categorias-proximos').on( 'click', 'button', function() {
```

```
        ordena(this, gridProximos);
    });

    $('#categorias-passados').on('click', 'button', function() {

        ordena(this, gridPassados);
    });

    function ordena(categorias, grid) {

        var sortValue = $(categorias).attr('data-sort-value');
        grid.isotope({ sortBy: sortValue });

        console.log(sortValue);
    }
});
```

```
/*--- formulario.js ---*/

$(document).ready(function() {

    var codigo = $('[name="codigo"]').val();
    var estudo = codigo.substr(2,1);
    var tarefa = codigo.substr(3,1);

    $('#nome').on('input', function() {

        var input = $(this);
        var valor = testarNome(input);

        validarCampo(input, valor);
    });

    $('#email').on('input', function() {

        var input = $(this);
        var valor = testarEmail(input);
```

```
        validarCampo(input, valor);
    });

    $('#cpf').on('input', function() {

        var input = $(this);

        if (estudo == 1) mascaraCpf(input);

        var valor = testarCpf(input);

        validarCampo(input, valor);

    });

    $('#numCartao').on('input', function() {

        var input = $(this);

        if (estudo == 1) mascaraNumCartao(input);

        var valor = testarNumCartao(input);

        validarCampo(input, valor);
    });

    $('#titularCartao').on('input', function() {

        var input = $(this);
        var is_name = input.val();

        validarCampo(input, is_name);
    });

    $('#validade').on('input', function() {

        var input = $(this);

        if(estudo == 1) mascaraValidade(input);
```

```
    var valor = testarValidade(input);

    validarCampo(input, valor);
});

$('#codigoSeguranca').on('input', function() {

    var input = $(this);
    var is_name = input.val();

    validarCampo(input, is_name);
});

$('#senha').on('input', function() {

    var input = $(this);
    var is_name = input.val();

    validarCampo(input, is_name);
});

$('#valor').on('change', function() {

    $(this).removeClass("invalido").addClass("valido");
})
})

function goBack() {

    window.history.back();
}

function limparCampos() {

    var campos = $(".formulario")
        .find('.campo')
        .serializeArray();
```

```
$.each(campos, function(i, campo) {

    var elemento = $('[name='+campo.name+']');
    elemento.removeClass("valido invalido");
    elemento.val("");
})
}



---




---


/*--- login.js ---*/

$(document).ready(function(){
    $(".social-login-box").height( $(".login-box").height() - 160 );
});



---




---


/*--- mascaras.js ---*/

function mascaraCpf(input) {

    var valor = input.val();

    input.attr("maxlength", "14");

    valor = valor.replace(/\D/g, "");
    valor = valor.replace(/(\d{3})(\d)/, "$1.$2");
    valor = valor.replace(/(\d{3})(\d)/, "$1.$2");
    valor = valor.replace(/(\d{3})(\d{1,2})$/, "$1-$2");

    input.val(valor);
}

function mascaraNumCartao(input) {

    var valor = input.val();

    input.attr("maxlength", "19");

    valor = valor.replace(/\D/g, "");
    valor = valor.replace(/(\d{4})(\d)/, "$1 $2");
    valor = valor.replace(/(\d{4})(\d)/, "$1 $2");
}
```



```
    valor = valor.replace(/(\d{4})(\d)/, "$1 $2");

    input.val(valor);
}

function mascaraValidade(input) {

    var valor = input.val();

    input.attr("maxlength", "7");

    valor = valor.replace(/\D/g, "");
    valor = valor.replace(/(\d{2})(\d)/, "$1/$2");

    input.val(valor);
}



---




---


/*--- migalha.js ---*/

$(document).ready(function() {

    var estudo = $('#estudo').text();
    var tarefa = $('#tarefa').text();

    console.log(estudo);
    console.log(tarefa);

    if (estudo == 'true') {

        $('#migalha').find('.ec0').addClass('btn-success').removeClass('disabled');

        switch (tarefa) {

            case '1':
                $('#migalha').find('.ec1.t1').addClass('btn-danger').removeClass('disabled');
                break;

            case '2':
                $('#migalha').find('.ec1.t1').addClass('btn-info').removeClass('disabled');
```

```
$('#migalha').find('.ec1.t2').addClass('btn-danger').removeClass('disabled');
break;

case '3':
$('#migalha').find('.ec1.t1').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.t2').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.t3').addClass('btn-danger').removeClass('disabled');
break;

case 'sus':
$('#migalha').find('.ec1.t1').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.t2').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.t3').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.sus').addClass('btn-danger').removeClass('disabled');
break;

case 'relatorio':
$('#migalha').find('.ec1.t1').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.t2').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.t3').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.sus').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.relatorio').addClass('btn-danger').removeClass('disabled');
break;
}

} else {

switch (tarefa) {

case '1':
$('#migalha').find('.ec0.t1').addClass('btn-danger').removeClass('disabled');
break;

case '2':
$('#migalha').find('.ec0.t1').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.t2').addClass('btn-danger').removeClass('disabled');
break;

case '3':
```

```
$('#migalha').find('.ec0.t1').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.t2').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.t3').addClass('btn-danger').removeClass('disabled');
break;

case 'sus':
$('#migalha').find('.ec0.t1').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.t2').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.t3').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.sus').addClass('btn-danger').removeClass('disabled');
break;

case 'relatorio':
$('#migalha').find('.ec0.t1').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.t2').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.t3').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.sus').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.relatorio').addClass('btn-danger').removeClass('disabled');
break;

case 'final':
$('#migalha').find('.ec1.t1').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.t2').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.t3').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.sus').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec1.relatorio').addClass('btn-info').removeClass('disabled');
$('#migalha').find('.ec0.t1').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.t2').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.t3').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.sus').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.ec0.relatorio').addClass('btn-success').removeClass('disabled');
$('#migalha').find('.final').addClass('btn-danger').removeClass('btn-light');
}
}
})
```

```
/*--- painel.js ---*/
```

```
(function(){
```

```

    'use strict';
    var $ = jQuery;
    $.fn.extend({
        filterTable: function(){
            return this.each(function(){
                $(this).on('keyup', function(e){
                    $('.filterTable_no_results').remove();
                    var $this = $(this),
                        search = $this.val().toLowerCase(),
                        target = $this.attr('data-filters'),
                        $target = $(target),
                        $rows = $target.find('tbody tr');

                    if(search == '') {
                        $rows.show();
                    } else {
                        $rows.each(function(){
                            var $this = $(this);
                            $this.text().toLowerCase().indexOf(search) === -1 ?
                                $this.hide() : $this.show();
                        })
                        if($target.find('tbody tr:visible').size() === 0) {
                            var col_count = $target.find('tr').first().find('td').size();
                            var no_results = $('<tr class="filterTable_no_results"><td
                                colspan="'+col_count+'">No results found</td></tr>');
                            $target.find('tbody').append(no_results);
                        }
                    }
                });
            });
        }
    });
    $(' [data-action="filter"]').filterTable();
})(jQuery);

$(function(){
    // attach table filter plugin to inputs
    $(' [data-action="filter"]').filterTable();
});

```

```
$('.container').on('click', '.panel-heading span.filter', function(e){
    var $this = $(this),
        $panel = $this.parents('.panel');

    $panel.find('.panel-body').slideToggle();
    if($this.css('display') != 'none') {
        $panel.find('.panel-body input').focus();
    }
});
 $('[data-toggle="tooltip"]').tooltip();
})
```

```
/*--- sus.js ---*/
```

```
$(document).ready(function() {

    $("#botao-enviar").click(function(e){

        var campos = $("#form-sus")
            .find('.form-check-input')
            .serializeArray();
        var error_free = true;
        var marcados = 0;

        $.each(campos, function(i, campo) {

            marcados++;

            var elemento = $('[name='+campo.name+']');
            var valido = elemento.hasClass("valido");
            var error_element = $("small", $(elemento).parent());

            console.log(elemento);

            if (!valido) {

                error_element.addClass("error").removeClass("no-error");
                error_free = false;
            }
        });
    });
});
```

```
    } else {  
  
        error_element.removeClass("error").addClass("no-error");  
    }  
});  
  
console.log(marcados);  
  
if (marcados < 10) {  
  
    e.preventDefault();  
    alert("Por favor, responda a todas as questões antes de prosseguir.");  
}  
});  
})
```

```
/*--- tarefa.js ---*/
```

```
$(document).ready(function() {  
  
    var idTarefa = $("#id-tarefa").text();  
    var idEvento = $('[name="id-evento"]').val();  
    var siglaEvento = $('[name="sigla"]').text();  
    var dataEvento = $('[name="data-evento"]').val();  
  
    var codigo = $('[name="codigo"]').val();  
    var estudo = codigo.substr(2,1);  
    var tarefa = codigo.substr(3,1);  
  
    // Tarefa 1  
    $.appear('#inscricao h5');  
  
    $('#inscricao h5').on('appear', function() {  
  
        if (tarefa == 1) {  
  
            if(siglaEvento == 'IHC') {  
  
                tarefaConcluida();  
  
            }  
  
        }  
  
    }  
});
```

```
    }
  }
});

$("#valor").on('click', function() {

  if (tarefa == 1) {

    if(siglaEvento == 'IHC') {

      tarefaConcluida();
    }
  }
});

//Tarefa 2
$("#botao-inscricao").click(function(e){

  var campos = $("#form-inscricao")
    .find('.ob')
    .serializeArray();
  var error_free = true;

  $.each(campos, function(i, campo) {

    var elemento = $(' [name='+campo.name+']');
    var valido = elemento.hasClass("valido");
    var error_element = $("small", $(elemento).parent());

    if (!valido) {

      error_element.addClass("error").removeClass("no-error");
      error_free = false;

    } else {

      error_element.removeClass("error").addClass("no-error");
    }
  })
});
```

```
    if (!error_free) {

        e.preventDefault();
        limparCampos();
        alert("Todos os campos devem ser preenchidos corretamente! Tente
            novamente.");

    } else {

        if (dataEvento < 0) {

            e.preventDefault();
            limparCampos();
            alert("Nao e possivel inscrever-se em um evento que ja aconteceu!");

        } else if (estudo == 1) {

            alert("Sua inscricao foi realizada com sucesso!");
        }

        if (tarefa == 2) {

            if(siglaEvento == 'SBQS') {

                $.get(jsRoutes.controllers.ControladorEstudos.fazerInscricao());
            }

            alert("Sua inscricao foi realizada com sucesso!");
        }
    }
});

//Tarefa 3
$("#botao-certificado").click(function(e){

    var campos = $("#form-certificado")
        .find('.ob')
        .serializeArray();
```



```
var error_free = true;

$.each(campos, function(i, campo) {

    var elemento = $('[name='+campo.name+']');
    var valido = elemento.hasClass("valido");
    var error_element = $("small", $(elemento).parent());

    console.log(dataEvento);

    if (!valido) {

        error_element.addClass("error").removeClass("no-error");
        error_free = false;

    } else {

        error_element.removeClass("error").addClass("no-error");
    }
})

if (!error_free) {

    e.preventDefault();
    limparCampos();
    alert("Todos os campos devem ser preenchidos corretamente! Tente novamente.");

} else if (dataEvento > 0) {

    e.preventDefault();
    alert("Nao e possivel emitir certificado de evento que ainda nao aconteceu!");

}
});

$(document).keypress(function(e){

    if (estudo == 0) {
```

```
        if (e.which == '13') e.preventDefault();
    }
});

function tarefaConcluida() {

    $.get(jsRoutes.controllers.ControladorEstudos.setConcluido(idTarefa));
}

});
```

```
/*--- validacao.js ---*/

function validarCampo(input, valor) {

    if(valor) {

        input.removeClass("invalido").addClass("valido");

    } else {

        input.removeClass("valido").addClass("invalido");
    }
}

function testarNome(input) {

    var regex = /[a-zA-Z]{3,}/;
    return regex.test(input.val());
}

function testarEmail(input) {

    var regex =
        /^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/;
    return regex.test(input.val());
}
```

```
function testarCpf(input) {  
  
    var regex = /^\d{3}\.\d{3}\.\d{3}-\d{2}$/;  
    return regex.test(input.val());  
}  
  
function testarNumCartao(input) {  
  
    var regex = /^\d/;  
    return regex.test(input.val());  
}  
  
function testarValidade(input) {  
  
    var regex = /^[0-1]\d\2\d{3}$/;  
    return regex.test(input.val());  
}  
  
function testarPreenchido(input) {  
  
    if (input != "") return true;  
    else return false;  
}
```
