

## Identificando Problemas que Impactam no Aprendizado de Uso do Simulador Educacional SSP–Edu

Leo Natan Paschoal<sup>1</sup>, Flávia de Souza Santos<sup>1</sup>, João Paulo Biazotto<sup>1</sup>,  
Ana Carolina Fernandes Spengler<sup>1</sup>, Renata Pontin de Mattos Fortes<sup>1</sup>

<sup>1</sup>Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo  
Caixa Postal 668 – 13.566-970 – São Carlos – SP – Brasil

{paschoalln, flaviasantos, joaobiazotto, ana.spengler}@usp.br

renata@icmc.usp.br

**Abstract.** *The CPU scheduling education through traditional pedagogical strategies is not trivial for an undergraduate student of Computing; hence the establishment of support mechanisms for teaching is encouraged. The SSP–Edu is a simulator that intends to support the CPU Scheduling Algorithm education and is available for open access. Previous studies describe that this simulator improves students’ knowledge, however, verifying the ease of learning the interface can be a factor that influences this process. In this sense, it is essential to check the usability problems of the simulator interface, which hinders the learnability. A method that aims to identify problems that cause difficulties in ease of use is the Cognitive Walkthrough. In this article, this method is used to inspect the SSP–Edu. From the inspection, 30 problems were identified, which need to be solved before this simulator is used in the classroom. In order to contribute to the theme, in the results, directions are suggested to promote improvements in the interface of this teaching support mechanism.*

**Resumo.** *O aprendizado de escalonamento de processos na CPU por meio de estratégias tradicionais de ensino não é trivial para um estudante de um curso de graduação em Computação, assim mecanismos de apoio ao ensino deste tópico é incentivado. Um simulador para apoiar especialmente o aprendizado do tópico “escalonamento de processos”, disponível gratuitamente, é o SSP–Edu. Estudos anteriores descrevem que esse simulador permite que os estudantes compreendam melhor o tópico, porém, verificar a facilidade de aprendizado da interface do simulador pode ser um fator que influencia esse processo. Nesse sentido, é importante verificar os problemas de usabilidade da interface do simulador, que prejudicam a facilidade de aprendizado de seu uso. Um método que visa identificar problemas que provocam dificuldades no aprendizado de uso é o Percurso Cognitivo. Neste artigo, esse método foi usado para inspecionar o simulador educacional SSP–Edu. A partir da inspeção, foram identificados 30 problemas, que precisam ser sanados antes desse simulador ser utilizado em aula. Visando contribuir com a temática, ao final, são discutidos direcionamentos para melhorias na interface desse mecanismo de apoio ao ensino, que facilitem seu aprendizado e favoreçam o uso de simuladores educacionais.*

### 1. Introdução

Os sistemas operacionais são responsáveis por gerenciar recursos do computador, como a unidade central de processamento (do inglês, *Central Process Unit* – CPU), a memória, os dispositivos de entrada e saída, e os sistemas de arquivos [Tanenbaum 2015]. Além

disso, têm a incumbência de fornecer uma interface entre o computador e o usuário [Tanenbaum 2015]. Em razão do sistema operacional ser representado como uma abstração do hardware para o software, compreender o que está acontecendo nos seus componentes é um dos principais desafios impostos durante o aprendizado da disciplina “Sistemas Operacionais” [Guimarães et al. 2016].

De modo geral, o ensino de cada recurso do sistema operacional abrange situações adversas. No que se refere à CPU, aprender sobre algoritmos de escalonamento de processos é dado como um desafio, uma vez que os alunos necessitarão compreender as características dos algoritmos, especificidades do sistema computacional, noções sobre preempção, e outros conceitos a serem estudados com maior profundidade ao longo da disciplina, quando os outros recursos do sistema operacional forem ensinados [Jbara 2017]. Além disso, os algoritmos de escalonamento possuem comportamentos que são difíceis de serem observados, dado que os mesmos sofrem alterações a partir dos valores de entrada de seus parâmetros [Killen et al. 2017]. Para minimizar essa dificuldade, professores dessas disciplinas têm buscado mecanismos de apoio ao ensino que possibilitem a visualização desses algoritmos, usando, por exemplo, simuladores educacionais.

Apesar de existirem diversos trabalhos sobre simuladores educacionais para apoiar o ensino de escalonamento de processos na CPU, até o momento não foram realizados esforços para avaliar a usabilidade desses mecanismos. No entanto, sabe-se que reduzir a frustração do aluno durante o uso de um simulador educacional é essencial para garantir o sucesso desse software [Van Nuland et al. 2017]. Além disso, se as interfaces do software exigirem que o aluno dispenda muito tempo aprendendo a utilizá-lo, o seu aprendizado do conteúdo poderá ser comprometido [Van Nuland et al. 2017]. Nesse sentido, a usabilidade do software é reconhecida como condição necessária para que o aluno alcance um aprendizado eficaz por intermédio de um software educacional [Tsironis et al. 2016].

Ao considerar que o objeto de análise é um simulador educacional, sua interface e elementos de interação tornam-se fundamentais, uma vez que, por meio deles, um novo usuário deve poder identificar os controles de entrada, componentes de navegação e informativos, dentre outros [Greis and Reategui 2010]. No contexto dos simuladores educacionais de escalonamento de processos na CPU, o usuário também precisará visualizar a transição de estados dos processos explorados durante uma simulação [Paschoal et al. 2019], bem como observar as informações sobre as propriedades desses processos [Killen et al. 2017], uma vez que contribuem para a compreensão do conteúdo estudado. Assim, proporcionar uma boa usabilidade da interface, com esses elementos de interação, torna-se essencial para auxiliar no processo de construção do conhecimento por parte dos estudantes.

Atualmente, existem simuladores educacionais dedicados a apoiar o aprendizado de escalonamento de processos na CPU; neste trabalho é considerado o SSP—Edu de [Paschoal et al. 2019], uma vez que abrange um conjunto amplo de algoritmos, permite a simulação de algoritmos preemptivos e não preemptivos, não tem limite quanto ao número máximo e mínimo de processos a serem criados e permite que o aluno defina novos processos durante a simulação do escalonamento de processos. O simulador possui licença aberta (*GNU General Public License*), o que possibilita o seu uso, modificação e replicação.

Visando contribuir com a temática de uso de simuladores para apoiar ensino de escalonamento de processos em CPU, este artigo descreve uma inspeção de usabilidade realizada para identificar problemas de usabilidade, que possam afetar o aprendizado do uso do simulador. Espera-se, com o estudo, oferecer subsídios à melhoria da usabilidade do

simulador educacional SSP—Edu, contribuindo com a divulgação desse tipo de software educacional e disseminação para mais usuários interessados.

Para abordar o estudo realizado, o artigo está organizado da seguinte forma. Na Seção 2 é apresentada uma visão geral sobre avaliação de usabilidade e o método Percurso Cognitivo. A Seção 3 descreve os materiais e métodos usados na avaliação da facilidade de aprendizado do sistema SSP—Edu. Na Seção 4, são consolidados os resultados. Por último, na Seção 5 são discutidas considerações e direções para trabalhos futuros.

## **2. Avaliação da facilidade de aprendizado no âmbito de usabilidade**

Na literatura é possível encontrar estudos sobre avaliação das interfaces de sistemas educacionais, buscando identificar evidências de problemas que podem afetar a usabilidade desses sistemas [Abreu et al. 2018, Moura et al. 2019]. Para avaliar a usabilidade de sistemas computacionais podem ser utilizados diferentes métodos, que segundo [Prates and Barbosa 2003] incluem a apreciação da interface (ou seu protótipo) e das interações, de maneira a qualificar a usabilidade.

A usabilidade é definida como “a medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso” [ISO 2018]. Ao mencionar a usabilidade de um sistema, [Dix et al. 2003] apresentam as três principais categorias dos princípios que podem ser aplicados no design de um sistema interativo para promover sua usabilidade: (a) facilidade de aprendizado, (b) flexibilidade e (c) robustez. A facilidade de aprendizado diz respeito ao grau de facilidade com que novos usuários podem iniciar uma interação efetiva com o sistema e alcançar um desempenho máximo. A flexibilidade corresponde a multiplicidade de maneiras pelas quais o usuário e o sistema trocam informações. Já a robustez, refere-se ao nível de apoio fornecido ao usuário na determinação do êxito e avaliação de metas.

Para avaliar-se usabilidade é necessário considerar os objetivos da avaliação, a disponibilidade de recursos (humanos, de tempo e financeiros) e a adequação do método. Os métodos de avaliação de usabilidade podem ser classificados em duas categorias: os baseados em usuários e os de inspeção. Os métodos de inspeção, em geral, fundamentam-se na compreensão dos usuários por meio de conhecimentos codificados em heurísticas, diretrizes, dados coletados remotamente ou modelos que preveem o desempenho dos usuários [Preece et al. 2015]. Os métodos de inspeção são aplicados por avaliadores, também chamados especialistas, que são profissionais experientes em usabilidade ou na área de conhecimentos/ conteúdos do software sob avaliação.

Dentre os métodos de avaliação por inspeção, o método Percurso Cognitivo prioriza a avaliação da facilidade de aprendizado de sistemas interativos. Assim, esse método foi escolhido para avaliar a “facilidade de aprendizado” do sistema SSP—Edu.

O Percurso Cognitivo (PC) proposto por [Lewis et al. 1990], fundamenta-se em uma análise do “passo a passo” que potenciais usuários realizariam para usar o sistema sob avaliação. A avaliação consiste, portanto, em observar o percurso que o usuário faria para alcançar seu objetivo durante a interação com o sistema. Esse método pode ser aplicado durante qualquer fase do desenvolvimento, para avaliar protótipos de baixa ou de alta fidelidade [Barbosa and Silva 2010]. Na Tabela 1, as etapas do processo de aplicação do PC, a serem realizadas pelo avaliador Especialista, são descritas.

Todas essas 4 etapas são essenciais para obter-se um resultado que represente efetivamente uma avaliação sobre a facilidade de aprendizado. A expectativa é que o avaliador isento percorra caminhos na interface buscando colocar-se no lugar do usuário e

Tabela 1. Etapas do PC

ETAPAS	Descrição
1. Preparação	Identificar os perfis de usuários, definir as tarefas a serem verificadas na avaliação, as ações necessárias para a realização de cada tarefa e preparar a interface, executável ou não, a ser inspecionada;
2. Coleta de dados e interpretação	Percorrer a interface de acordo com a sequência de ações definidas como necessárias para a realização de cada tarefa, e para cada ação, analisar como o usuário a executaria, respondendo e justificando às 4 questões fundamentais: <b>Q1.</b> <i>O usuário tentaria atingir o efeito correto?</i> <b>Q2.</b> <i>O usuário perceberia que a ação correta está disponível?</i> <b>Q3.</b> <i>O usuário conseguiria associar a ação correta com o efeito que está tentando atingir?</i> <b>Q4.</b> <i>Caso a ação correta seja realizada, o usuário perceberia que está progredindo para concluir a tarefa?</i>
3. Consolidação dos resultados	Sintetizar os resultados da avaliação, ressaltando o que o usuário deve saber a priori para realizar as tarefas, e o que deve aprender enquanto as realiza;
4. Relato dos resultados	Elaborar um relatório consolidado com os problemas encontrados e sugestões de correção.

verificando se há algum problema potencial ou tentando prever quais as dificuldades de interação. Na próxima seção são descritas as quatro etapas, conforme foram aplicadas, do PC para avaliação da facilidade de aprendizado do SSP—Edu.

### 3. Materiais e Métodos

#### 3.1. Preparação

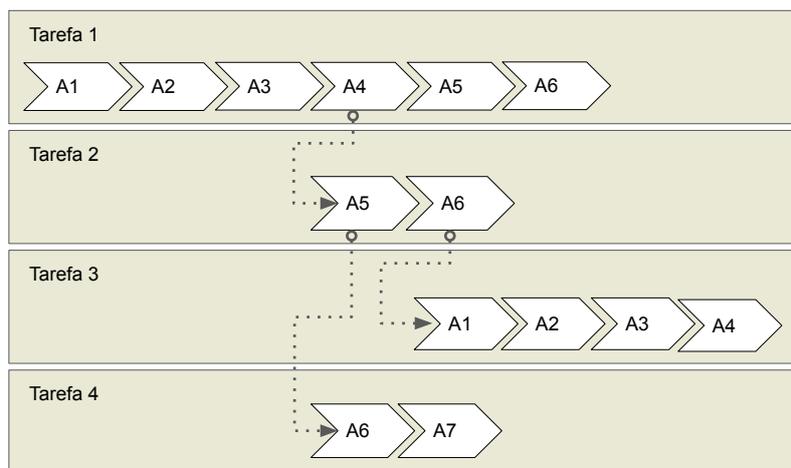
Inicialmente, o responsável pela avaliação organizou os objetos de estudo, definindo a lista de tarefas investigadas e a sequência de ações esperadas para a realização de cada tarefa. Foi preparado o material de apoio, contendo a lista de perguntas do método e a descrição do perfil de usuários (descrição dos alunos de uma disciplina de Sistemas Operacionais). Essa avaliação foi iniciada em junho de 2018, e teve duração de uma semana.

A seguir estão descritas as quatro tarefas que foram definidas, por representarem as principais finalidades do SSP—Edu. Além disso, essas tarefas cobriam todas as funcionalidades do simulador SSP—Edu. Na Figura 1, são esquematizadas as sequências de ações necessárias para realiza-las no sistema (A1). As tarefas:

- T1** - Crie uma simulação de um sistema computacional do tipo *batch*, que execute qualquer um dos algoritmos disponíveis no simulador, com quantos processos desejar.
- T2** - Crie uma simulação que suporta a execução do algoritmo *Shortest Remaining Time Next* e possua pelo menos três processos antes de começar a execução. Visualize as informações sobre os processos.
- T3** - Execute a simulação que configurou na **T2**. Crie um novo processo durante a execução de qualquer um dos processos.
- T4** - Crie uma simulação que suporte a execução do algoritmo *Round-Robin* e possua pelo menos três processos antes de começar a execução. Do total de processos criados, pelo menos um deve ser do tipo *I/O Bound*. Escalonar os demais processos.

#### 3.2. Coleta de dados e interpretação

Como a aplicação do método PC prevê a participação de especialistas, foi definido como critério de seleção que os avaliadores deveriam ser especialistas em Interação Humano-Computador, com experiência em avaliações de usabilidade mínima de 2 anos, e com conhecimento prévio sobre a aplicação do método PC, selecionado para este estudo.



**Figura 1. PC no Simulador SSP—Edu: Tarefas e respectivas Ações**

Foram enviados *e-mails* convidando doutorandos de dois Programas de Pós-Graduação em Computação, que já pesquisam e possuem experiência na avaliação de sistemas computacionais. Três aceitaram o convite, e se disponibilizaram a participar da avaliação proposta. Os especialistas que se voluntariaram receberam e assinaram o Termo de Consentimento Livre e Esclarecido (TCLE), esclarecendo ao participante sobre o uso dos dados fornecidos e sobre a possibilidade de desistir de participar da avaliação do sistema a qualquer momento, sem prejuízo próprio.

De maneira a conduzir a avaliação por meio da aplicação do método PC, houve primeiramente a aplicação de um estudo-piloto com as 4 tarefas e considerando-se as respectivas ações definidas. Esse piloto foi realizado por um aluno de mestrado, com experiência profissional em avaliação de interfaces. Durante o piloto, foi solicitado que o avaliador seguisse o roteiro de aplicação do método e fizesse sugestões de melhorias no roteiro. Dentre as melhorias sugeridas, foi recomendado que a descrição das ações de cada tarefa fossem melhor detalhadas, correspondendo a interações na interface. Após esse estudo-piloto, foram realizadas modificações conforme sugeridas.

Na fase de coleta de dados, os especialistas foram convidados a irem individualmente até o local em que a inspeção seria realizada, de acordo com os horários de sua conveniência. Um computador foi configurado para realização da avaliação. Foi solicitado que o especialista preenchesse o TCLE e um formulário de caracterização de sujeitos, lhe foi entregue o roteiro de avaliação. Foram disponibilizados dois monitores para a realização da avaliação, e os especialistas foram instruídos a utilizar um deles para acessar o simulador SSP—Edu e o outro para ver o material de apoio, contendo o conjunto de tarefas, as ações e as quatro questões do PC a serem respondidas. Antes de iniciar a avaliação, foi esclarecido aos especialistas que poderiam manifestar possíveis dúvidas sobre o procedimento, mesmo durante sua realização.

#### **4. Resultados**

Os especialistas realizaram as tarefas e respectivas ações descritas no protocolo do método. Assim, eles puderam identificar os possíveis problemas que os usuários encontrariam na interação com o sistema. Os 30 problemas encontrados durante a realização das ações para efetivar as tarefas estão descritos na Tabela 2, e estão agrupados nas 4 tarefas definidas. A primeira coluna da Tabela 2 é a Ação numerada (A1 - A7), na qual o problema foi detectado.

Na segunda coluna são atribuídos identificadores (IDs) aos problemas encontrados, de maneira a facilitar sua identificação posterior. Na coluna ‘Descrição do problema’, tem-se a tarefa na qual o problema foi percebido e a descrição dos problemas identificados. Na última coluna, estão indicados os especialistas que apontaram cada um desses problemas.

**Tabela 2. Lista dos problemas encontrados pelos especialistas**

Ação	ID	Descrição do problema	E <sub>n</sub>
<b>Tarefa 1 (T1)</b>			
A1	1	A nomenclatura do botão “Iniciar simulação” pode gerar confusão, pois o usuário ainda não definiu nada, e portanto a semântica desse botão não é óbvia	E1
	2	A posição do botão “Iniciar simulação” que inicia a simulação está abaixo de um longo texto	E2
A2	3	O usuário pode não identificar as opções de simulação disponíveis	E2
	4	O usuário pode não perceber que está progredindo na tarefa	E2
A3	5	Falta uma instrução para a próxima ação, e pode demandar sobrecarga cognitiva do usuário	E1 E2 E3
	6	É necessária mais uma ação caso o usuário deseje visualizar descrição do algoritmo	E2
A4	7	O tamanho da fonte na mensagem quanto às propriedades do processo está pequeno	E1
	8	O <i>range slider</i> parece ser divisão de página e não um elemento clicável	E1 E2 E3
	9	A mensagem “Importante: Você não tem nenhum processo criado” logo no início pode confundir ou inibir o usuário	E3
A5	10	São exibidas muitas opções, o usuário pode perder o foco da tarefa	E1 E2
	11	O usuário pode ter dúvidas sobre seu progresso quanto a criação do processo	E2
	12	Falta um botão de voltar para página anterior	E2
A6	13	O botão “Avançar” em posição ruim	E1
	14	Falta <i>feedback</i> sobre a transição de processos	E1
	15	A página de escalonamento precisa ser rolada para que o botão “Finalizar” seja visualizado	E1
	16	O usuário pode não se lembrar das opções que selecionou anteriormente, não compreendendo o comportamento do escalonador	E2
<b>Tarefa 2 (T2)</b>			
A5	17	Falta <i>feedback</i> para informar que o processo foi criado	E1
	18	Falta <i>feedback</i> com número de processos criados	E1
A6	19	O texto do botão “Verificar informações sobre os processos criados” pode causar confusão no usuário, uma vez que o usuário pretendia visualizar	E2
	20	Não fica claro que os processos criados estão disponíveis para visualização	E2
<b>Tarefa 3 (T3)</b>			
A1	21	Falta apresentar a visualização dos processos criados antes de sua simulação	E3
A2	22	O texto do botão “Avançar” não remete a ação a ser executada	E1 E2 E3
	23	Falta explicação clara do que fazer para execução da ação	E2
	24	Se o usuário não tem conhecimento prévio sobre os conceitos inerentes ao tópico, ele pode não saber qual o efeito esperado	E2
	25	Não é possível vislumbrar quantos passos ainda são esperados para finalizar a execução da simulação	E2
	26	Sem padronização no posicionamento do botão “Avançar”	E3
A3	27	Botão “Criar um novo processo” mal posicionado	E1 E2
A4	28	Botão “Avançar” mal posicionado	E1 E2
<b>Tarefa 4 (T4)</b>			
A7	29	O usuário só tem acesso à opção de imprimir ( <i>link</i> “Preparar página para impressão”) após finalizar a simulação	E1
	30	O <i>link</i> para imprimir pode não ser identificado devido a falta de familiaridade	E2 E3

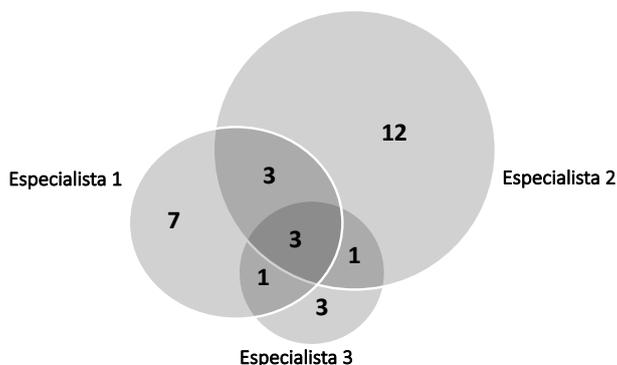
Para proceder as análises, com base nos problemas identificados pelos especialistas (E1, E2 ou E3) e associados às tarefas, obteve-se:

- na (T1), foram encontrados 16 problemas, 2 deles foram apontados pelos três especialistas, 1 deles por dois especialistas e os outros por apenas um especialista.
- na (T2), como as ações de 1 a 4 são as mesmas de T1, os problemas destas ações em específico são os mesmos aos encontrados na (T1) (9 problemas nas ações de 1 a 4), e nas ações 5 (A5) e 6 (A6), inexistentes na (T1), foram encontrados outros 4 problemas, sendo 1 deles visto por dois especialistas e os outros 3, apenas por um dos especialistas. Assim, a quantidade de problemas na (T2) totalizou 13.
- na (T3), com 4 ações, foram encontrados 8 problemas, sendo 1 deles indicado pelos 3 especialistas e 2 deles por dois especialistas.
- na (T4), as ações de 1 a 6 já haviam sido avaliadas nas tarefas anteriores (T1 e T2 - 13 problemas), e em relação à ação 7, foram indicados outros 2 problemas, 1 deles por dois dos especialistas. Assim, a quantidade de problemas na (T4) totalizou 15.

A partir da identificação dos 72 problemas (30 problemas se forem desconsideradas as repetições das ações), foi realizada uma análise com a intenção de detectar os problemas mais recorrentes (*i.e.*, observados por todos os especialistas).

No diagrama de Venn da Figura 2, tem-se a representação das quantidades dos problemas distintos entre si, reportados por cada um dos especialistas. Foram encontrados, ao todo, 30 problemas diferentes entre si, sendo que dentre eles, o E1 encontrou 14 problemas, o E2, 19 problemas e o E3, 8 problemas. Como determinadas ações são comuns a mais de uma tarefa, podemos supor que os problemas estariam presentes no decorrer em todas as ações comuns. Desse modo, tem-se que, considerando-se todas as ações avaliadas nas 4 tarefas, o E1 identificou 24 problemas, o E2 identificou 33 problemas, e o E3, 15 problemas.

Desses problemas, 3 foram encontrados por todos os especialistas (problemas ID = 5, ID = 8 e ID = 22 apresentados na Tabela 2), e a maioria dos outros problemas foram encontrados por somente 1 dos especialistas. Argumenta-se que esses 3 problemas, identificados em comum pelos três especialistas, podem ser caracterizados como mais sugestivos a prejudicar o uso do SSP—Edu. Assim como ocorreu com os especialistas, é muito provável que os alunos que utilizam o SSP—Edu também fiquem mais sujeitos a se depararem com esses problemas, durante a utilização do sistema.



**Figura 2. Conjuntos de problemas encontrados pelos especialistas**

De maneira a propiciar uma classificação dos problemas encontrados durante a aplicação do PC, foram consideradas as 10 heurísticas definidas por [Nielsen 1994b] para

avaliar a usabilidade de sistemas computacionais. A classificação pode ser vista na Tabela 3. A maioria dos problemas se enquadra na “H1 - Visibilidade do estado do sistema”, o que pode indicar a necessidade de rever os “*feedbacks*” apresentados pelo SSP—Edu. Vale salientar que nenhum dos problemas relatados pelos Especialistas foi classificado para as heurísticas H5, H9 e H10.

**Tabela 3. Mapeamento dos problemas nas 10 Heurísticas de Nielsen**

#	Heurística	IDs dos problemas
H1	Visibilidade do estado do sistema	4, 6, 9, 11, 14, 16, 17, 18, 21 e 25
H2	Compatibilidade entre o sistema e o mundo real	1, 3, 19, 20, 22 e 30
H3	Controle e liberdade do usuário	12
H4	Consistência e padrões	8, 13, 26, 27 e 28
H5	Prevenção de erro	-
H6	Reconhecimento em vez de lembrança	5, 23 e 24
H7	Flexibilidade e eficiência de uso	-
H8	Estética e design minimalista	2, 7, 10, 15 e 29
H9	Ajudar a reconhecer, diagnosticar, e recuperar de erros	-
H10	Ajuda e documentação	-

Para cada problema também foi atribuído um grau de severidade (Tabela 4), como proposto por [Nielsen 1994a], complementando a avaliação por meio do PC. Nessa análise, foi observado apenas um problema com grau de severidade 4, cuja detecção foi feita por todos os especialistas que aplicaram o método PC. Esse problema (ID 8), é mostrado na Figura 3. O *range slider* parece ser um elemento que divide a página do simulador, e não um elemento a ser escolhido. Os problemas IDs 5 e 22 também foram detectados por todos os especialistas durante o PC, e juntamente com outros 11 problemas receberam grau de severidade 3.



**Figura 3. Captura de tela da interface do simulador SSP—Edu**

Os problemas encontrados podem dificultar o aprendizado do usuário sendo uma barreira na interação e no aprendizado de uso do sistema SSP—Edu. Os especialistas indicaram sugestões para corrigi-los. Quatro exemplos de correções sugeridas durante a aplicação do PC são apresentados a seguir, juntamente com o ID do problema dos quais elas tratam:

**Tabela 4. Atribuição de graus de severidade aos problemas**

#	Severidade	IDs dos problemas
0	Não é problema	-
1	Cosméticos	7
2	Baixa prioridade	1, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 26, 27, 28 e 29
3	Alta prioridade	2, 3, 4, 5, 6, 15, 16, 21, 22, 23, 24, 25 e 30
4	Catastróficos	8

- Problema ID 5: Apresentar instrução para escolher somente um dos algoritmos de escalonamento;
- Problema ID 8: Alterar a aparência do *range slider*;
- Problema ID 22: Alterar texto do botão.
- Problemas IDs 23 e 24: Colocar mensagem de explicação

## 5. Conclusões

Este artigo teve o objetivo de identificar problemas que impactassem o aprendizado de uso de um Simulador Educacional, o qual já apresentava resultados favoráveis quanto a melhoria de conhecimentos sobre escalonamento de processos, por parte de estudantes. A motivação para este trabalho foi a de disseminar a novos estudantes, de modo que o referido sistema educacional possuísse boa usabilidade, em termos da facilidade de aprendizado da interface do sistema. Para tanto, foi utilizado o método PC.

Foi definido um protocolo de avaliação para condução do PC, bem como foram convidados especialistas em avaliação de usabilidade de software, todos pós-graduandos nessa área. A partir da inspeção, com 3 especialistas, foram identificados 30 problemas, que potencialmente afetam o aprendizado do usuário durante o uso do simulador educacional. Esses problemas proporcionaram mais análises, as quais derivaram um conjunto de soluções, proposto pelos especialistas.

Por meio da inspeção de usabilidade realizada, com os problemas identificados e respectivas soluções, como trabalho futuro, espera-se que a manutenção do simulador seja realizada. Além disso, espera-se que o estudo experimental descrito no trabalho de [Paschoal et al. 2019] seja replicado, uma vez que o experimento foi executado na versão do simulador que possui os problemas relatados neste artigo, e que potencialmente atrapalham o aprendizado do usuário. Acredita-se que a replicação do experimento conduzida com o simulador, no qual tenham sido corrigidos os problemas reportados, possa demonstrar com mais eficácia o quanto o simulador apoia efetivamente o aprendizado sobre escalonamento de processos, por mais alunos.

## Agradecimentos

Os autores gostariam de agradecer a CAPES - Código de Financiamento 001, ao CNPq.

## Referências

- (2018). ISO 9241-11: 2018 Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts.
- Abreu, C. A., Rosa, J. C. S., and Souza Matos, E. (2018). Usabilidade de aplicativos móveis educacionais infantis: design e avaliação de interação do *fantastic pirates*. *RENOTE*, 16(1).

- Barbosa, S. and Silva, B. (2010). *Interação Humano-Computador*. Elsevier, Rio de Janeiro.
- Dix, A., Finlay, J., Abowd, G. D., and Beale, R. (2003). *Human Computer Interaction*. Pearson Education Limited, Harlow, England, 3rd edition.
- Greis, L. K. and Reategui, E. (2010). Um simulador educacional para disciplina de física em mundos virtuais. *RENOTE*, 8(2):1–10.
- Guimarães, M. P., Scamati, V., Neto, M. P., Martins, V. F., Dias, D. R. C., and Brega, J. R. F. (2016). A process-scheduling simulator based on virtual reality technology. In *International Conference of Computer Systems and Applications*, pages 1–6.
- Jbara, Y. H. (2017). A new visual tool to improve the effectiveness of teaching and learning CPU scheduling algorithms. In *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies*, pages 1–6.
- Killen, S., Giese, E., Huynh, H., and Indratmo (2017). Marble MLFQ: An educational visualization tool for the multilevel feedback queue algorithm. In *IEEE Annual Information Technology, Electronics and Mobile Communication Conference*, pages 663–669.
- Lewis, C., Polson, P. G., Wharton, C., and Rieman, J. (1990). Testing a Walkthrough Methodology for Theory-based Design of Walk-up-and-use Interfaces. In *Conference on Human Factors in Computing Systems*, pages 235–242.
- Moura, M. P. C., Nunes, R. P., Mocbel, M. Â. R., and de Souza Farias, F. (2019). Protótipo de Aplicativo Educativo para o ensino de POO: Avaliação da Usabilidade e Experiência do Usuário. *RENOTE*, 17(3):305–314.
- Nielsen, J. (1994a). Heuristic evaluation. In *Usability inspection methods*, pages 25–62. John Wiley & Sons, Inc.
- Nielsen, J. (1994b). *Usability engineering*. Morgan Kaufmann.
- Paschoal, L. N., Biazotto, J. P., Spengler, A. C. F., Oliveira, M. M., Felizardo, K., and Nakagawa, E. Y. (2019). Towards an educational simulator to support the CPU scheduling algorithm education. In *International Symposium on Computers in Education*, pages 1–6.
- Prates, R. O. and Barbosa, S. D. J. (2003). Avaliação de interfaces de usuário—conceitos e métodos. In *Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação, Capítulo*, volume 6, page 28.
- Preece, J., Rogers, Y., and Sharp, H. (2015). *Interaction Design: Beyond Human–Computer Interaction*. John Wiley & Sons, West Sussex, United Kingdom, 4th edition.
- Tanenbaum, A. (2015). *Modern Operating Systems*. Prentice Hall Press, Upper Saddle River, NJ, USA, 4rd edition.
- Tsironis, A., Katsanos, C., and Xenos, M. (2016). Comparative usability evaluation of three popular mooc platforms. In *IEEE Global Engineering Education Conference*, pages 608–612.
- Van Nuland, S. E., Eagleson, R., and Rogers, K. A. (2017). Educational software usability: Artifact or design? *Anatomical Sciences Education*, 10(2):190–199.