



**UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO**  
**UNIDADE ACADÊMICA DE SERRA TALHADA**  
**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**WILLIANY THALITA ALMEIDA VERAS**

**Desenvolvimento e Avaliação de Usabilidade do  
Jogo Educacional Desktop JAavatar para o  
Ensino da Programação Orientada a Objetos:  
uma abordagem gamificada**

Serra Talhada,  
Março/2021

**Williany Thalita Almeida Veras**

**Desenvolvimento e Avaliação de Usabilidade do  
Jogo Educacional Desktop JAvatar para o  
Ensino da Programação Orientada a Objetos:  
uma abordagem gamificada**

Trabalho de Conclusão de Curso apresentada ao Curso de Bacharelado em Sistemas de Informação da Unidade Acadêmica de Serra Talhada da Universidade Federal Rural de Pernambuco como requisito parcial à obtenção do grau de Bacharel.

Orientador: Prof. Dr. Richarlyson Alves D'Emery

Serra Talhada,  
Março/2021

Dados Internacionais de Catalogação na Publicação  
Universidade Federal Rural de Pernambuco  
Sistema Integrado de Bibliotecas  
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

---

V476d

Veras, Williany Thalita Almeida

Desenvolvimento e Avaliação de Usabilidade do Jogo Educacional Desktop JAvatar para o Ensino da Programação Orientada a Objetos: uma abordagem gamificada / Williany Thalita Almeida Veras. - 2021.  
27 f. : il.

Orientador: Richarlyson Alves DEmerly.  
Inclui referências.

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Bacharelado em Sistemas da Informação, Serra Talhada, 2021.

1. Jogos Educacionais. 2. Gamificação. 3. Ensino de Programação. 4. Avaliação de Usabilidade. I. DEmerly, Richarlyson Alves, orient. II. Título

CDD 004

---

**UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO**  
**UNIDADE ACADÊMICA DE SERRA TALHADA**  
**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**WILLIANY THALITA ALMEIDA VERAS**

**Desenvolvimento e Avaliação de Usabilidade do Jogo Educacional Desktop  
JAvatar para o Ensino da Programação Orientada a Objetos: uma abordagem  
gamificada**

Trabalho de Conclusão de Curso julgado adequado para obtenção do título de Bacharel em  
Sistemas de Informação, defendida e aprovada por unanimidade em 02/03/2021 pela banca  
examinadora.

Banca Examinadora:

Prof. Dr. Richarlyson Alves D'Emery  
Orientador  
Universidade Federal Rural de Pernambuco

Prof. Me. Héldon Oliveira Albuquerque  
Universidade Federal Rural de Pernambuco

Prof. Me. Carlos André Batista  
Universidade Federal Rural de Pernambuco

# DEDICATÓRIA

Dedico este trabalho, a minha mãe, Terezinha Almeida dos Santos, que me ensinou o poder da educação desde criança, me incentivando a estudar e alcançar meus objetivos. Apoiando-me e confortando, nunca mediu esforços para que eu pudesse alcançar meus sonhos e chegar até esta nova fase da minha vida.

# AGRADECIMENTOS

Agradeço a Deus, que iluminou meu caminho, sempre me acalmando e me dando forças para continuar persistindo.

Agradeço a toda a minha família, especialmente a minha mãe, Terezinha Almeida, por todo o esforço investido na minha educação. Ao meu pai Willamys Veras e ao meu padrasto Alexandre Menezes, por me apoiarem incondicionalmente. A minha irmã, Khaleesi Almeida, por sua alegria e carinho, agradeço a Deus por ter me presenteado com uma pequena rainha.

Aos meus amigos, Elvis Nogueira e João Emerson, que me deram muito força e incentivo, nunca me deixaram fraquejar. Essa conquista seria impossível sem o apoio deles.

Ao meu orientador Prof. Dr. Richarlyson Alves D'Emery, por ter me acompanhado nessa trajetória, sempre indicando o melhor caminho para chegar ao final deste trabalho. Seus conhecimentos foram essenciais na realização desta pesquisa.

Aos professores e todos os funcionários da UAST-UFRPE, que mesmo durante a pandemia da COVID-19, trabalharam duro para que nós alunos pudéssemos continuar aprendendo e evoluindo, agradeço profundamente pelos vossos profissionalismos e esforços.

A todos, o meu muito, obrigada.

*“É muito melhor adaptar a tecnologia ao  
usuário do que forçar o usuário a se  
adaptar à tecnologia”*

- Larry Marine

.

# SUMÁRIO

1	INTRODUÇÃO	08
2	REFERENCIAL TEÓRICO	09
2.1	Softwares Educacionais	09
2.2	Usabilidade de Software	09
2.3	Avaliação de Software	10
2.4	Trabalhos Relacionados	12
2.4.1	Análise Comparativa	14
3	MÉTODOS	15
3.1	O Jogo JAvatar	15
3.1.1	Conceito	15
3.1.2	Desenvolvimento	15
3.1.3	Regras	16
3.2	Avaliação de JAvatar	16
4	RESULTADOS E DISCUSSÕES	19
4.1	Elementos Visuais e Interativos de JAvatar	19
4.2	Resultado da Avaliação	21
4.3	Análise e Interpretação da Avaliação de JAvatar	22
5	CONCLUSÃO	24
	REFERÊNCIAS	25



# Desenvolvimento e Avaliação de Usabilidade do Jogo Educacional Desktop JAvatar para o Ensino da Programação Orientada a Objetos: uma abordagem gamificada

Williany Thalita A. Veras<sup>1</sup>, Richarlyson A. D'Emery<sup>1</sup>

<sup>1</sup>Universidade Federal Rural de Pernambuco (UFRPE),  
Caixa Postal 063 – 56.909-535 – Serra Talhada – PE – Brasil

{williany.veras, ricodemery}@gmail.com

**Abstract. Introduction:** *The teaching process demands solutions that encourage student participation and interest. Digital Games afford a higher level of motivation and interactivity. Goals:* Present the JAvatar game and your usability assessment. **Methods:** *Java, XML and architectural patterns are used with iterative and incremental development. The usability assessment uses prospective technique by questionnaires with analysis by Chi-Square, ANOVA and System Usability Scale. Result:* An educational game is presented that provides significant contributions to the teaching-learning process of Object Oriented Programming and your design was classified as excellent.

**Resumo. Introdução:** *O processo de ensino demanda soluções que estimule a participação e interesse de alunos. Jogos digitais são exemplos que proporcionam maior nível de motivação e interatividade. Objetivo:* Apresentar o jogo JAvatar e sua avaliação de usabilidade. **Métodos:** *Utiliza-se Java, XML e padrões de arquitetura de software com desenvolvimento iterativo e incremental. A avaliação de usabilidade utiliza técnica prospectiva por questionários com análise por Qui-Quadrado, ANOVA e System Usability Scale. Resultado:* Apresenta-se um jogo educacional que fornece contribuições significantes para o processo de ensino-aprendizagem de Programação Orientada a Objetos e design classificado como excelente.

## 1. Introdução

Em meio aos progressos da Computação, o uso de tecnologias em sala de aula vem transformando os padrões de ensino e de aprendizagem. A utilização de jogos para fins educativos é cada vez mais empregada no ambiente acadêmico, pois tornam a aprendizagem mais atrativa. Segundo [Clark e Mayer 2012], programar é uma tarefa desafiadora e complexa, logo, é necessário fazer uso da própria tecnologia para descobrir novas formas que facilitem o aprendizado e o ensino da Programação.

Nesse cenário, jogos que proporcionam um maior nível de interatividade com usuário são necessários, porque se tornam mais atrativos, por comumente empregar conceitos de usabilidade, sendo este um termo usado para classificar sistemas fáceis de aprender e usar. Em outras palavras, a usabilidade diz quão intuitiva é a interface gráfica, característica pela qual o usuário expressa interesse ou não em usufruir o sistema [Filho 2008].

Uma problemática conhecida pelos profissionais de Educação em Computação são os altos índices de reprovação nas disciplinas de Programação, uma vez que vários

alunos têm dificuldades em entender os conceitos básicos para aplicá-los nas construções de sistemas [Henrique e Rebouças 2015].

Segundo [Mombach et al. 2018], Programação, em especial a Programação Orientada a Objetos (POO), tem uma grande quantidade de conceitos e muitos alunos possuem dificuldade na memorização dos mesmos. Porém, com a crescente demanda de desenvolvedores ao redor do mundo, esta dificuldade se torna relevante. Logo, impulsiona-se o desenvolvimento desta pesquisa.

Sendo assim, este artigo apresenta o jogo educacional JAvatar para auxiliar no processo de ensino-aprendizagem de POO para alunos de cursos que abordam POO, bem como sua avaliação de usabilidade no contexto pedagógico e design.

Este documento segue estruturado da seguinte forma: na Seção 2 apresentam-se os referenciais teóricos e trabalhos relacionados. Na Seção 3 têm-se os métodos, enquanto na Seção 4 são apresentados os resultados e discussões. Na Seção 5 são tecidas as conclusões. Por fim, as Referências que guiaram a pesquisa.

## **2. Referencial Teórico**

### **2.1. Softwares Educacionais**

*Softwares* educacionais são ferramentas que auxiliam alunos no processo de ensino-aprendizagem por oferecer uma apresentação visual do conhecimento e possibilitar estímulo e foco. Porém, não só visam auxiliar o ensino, mas também ajudar professores e promover mais recursos em salas de aulas. Segundo [Morais 2003], quando utilizado de forma adequada, *software* educacional pode gerar consequências fantásticas, como habilidades de resolver problemas e de investigação, aproximando a prática da teoria.

Jogos são instrumentos efetivos no processo de aprendizagem e ensino, pois são alternativas viáveis para tornar as aulas mais interessantes e diminuir o nível de abstração advindos da Programação [Henrique e Rebouças 2015]. Para [Piteira e Haddad 2011], *softwares* educativos proporcionam diversão e motivam os estudantes a aprenderem, assim facilitando o processo de fixação e aprendizagem do conteúdo.

### **2.2. Usabilidade de Software**

A usabilidade é um atributo de qualidade, que foca em princípios de boas práticas para avaliar a facilidade e medição dos esforços necessários, para que o usuário tenha sucesso ao executar suas tarefas. Segundo [Cybis, Betiol e Faust 2010], a usabilidade é o acordo entre usuário, interface, ambiente e a tarefa a ser executada.

[Preece, Rogers e Sharp 2007] aponta que a usabilidade deve assegurar que a experiência com os produtos seja fácil, eficiente e agradável, tendo como meta a busca pela eficiência, segurança, utilidade, capacidade de aprendizagem e memorabilidade. Complementando, a norma ISO 9241-11 [ABNT 2002] aponta que a usabilidade é a forma com que um produto possa ser utilizado em um contexto específico e por usuários específicos, para atingir objetivos com satisfação, eficácia e eficiência. E de acordo com o padrão ISO/IEC 9126 [ISO 2001], a usabilidade é um conjunto de atributos que dão ênfase no esforço necessário para que um produto possa ser utilizado.

Nota-se que à usabilidade possui diversas definições, porém, pode-se dizer que é uma forma de deixar o sistema mais fácil, agradável e eficiente, assim, seu uso é

necessário, pois é uma correlação benéfica entre usuários e sistemas, podendo vir a ser um fator decisivo na escolha de um *software*. Entretanto, para alcançar uma interface que favoreça essa relação, [Cybis, Betiol e Faust 2010] dizem ser necessário soluções que respeitem certos critérios, princípios ou heurísticas de usabilidade.

### 2.3. Avaliação de Software

Avaliação de *software* é de vital importância no processo de desenvolvimento por ajudar a identificar problemas de interação, tornando o sistema mais utilizável e de acordo com o que os usuários desejam. Essas avaliações são realizadas através de métodos que mostram um caminho mais eficaz para o aperfeiçoamento da usabilidade de um sistema. Porém, vários modelos de usabilidade foram desenvolvidos ao longo dos anos. A Tabela 1 apresenta alguns métodos de usabilidade, assim como suas definições.

O modelo de Nielsen [Nielsen 1993, Nielsen 1994] é um método de verificação que seguem um grupo (heurísticas) de regras de usabilidade, as quais testam os usuários mediante interfaces, com a finalidade de localizar falhas de usabilidade.

Idealmente, a avaliação deve ocorrer durante todo o processo de desenvolvimento do sistema, desde sua prototipação até a fase final, pois os resultados das avaliações permitem melhorar as interfaces [Rocha e Baranaukas 2003].

Em usabilidade, uma avaliação heurística é uma técnica rápida e barata de melhorar um sistema. [Nielsen e Molich 1990] desenvolveram um conjunto de heurísticas, essas regras buscam revisar a interface, dando importância à experiência do usuário. Essas regras, posteriormente, ficaram mundialmente conhecidas como as Heurísticas de Nielsen [Nielsen, 1994], a saber: (i) visibilidade do estado do sistema, o sistema deve manter os usuários informados através do *feedback* certo no momento apropriado; (ii) equivalência entre o sistema e o mundo real, o *software* deve sempre falar a língua dos usuários; (iii) liberdade e controle do usuário, sistemas devem oferecer soluções de saídas de emergência, voltar, desfazer e editar; (iv) consistência e padrões, o sistema deve manter à consistência e o padrão visual e de linguagem; (v) prevenção de erro, o sistema deverá ter um bom *design* para que possa prevenir esses erros; (vi) reconhecer ao invés de lembrar, utilizar elementos já conhecidos por grande parte dos usuários; (vii) flexibilidade e eficiência de uso, para os usuários avançados o sistema pode ser rápido e ser fácil para os usuários mais leigos; (viii) estética e *design* minimalista, a página deverá conter somente aquilo que é necessário; (ix) auxiliar usuários a reconhecer, diagnosticar e recuperar-se de ações erradas; as mensagens de erro devem ser simples e diretas; e (x) ajuda e documentação, os itens de auxílio devem estar ao alcance do usuário.

[Brooke 1996] apresenta *System Usability Scale* (SUS) como método de verificação de qualidade de *software* que pode ser usada para validar várias aplicações e interfaces. Utiliza um questionário com 10 questões, alternadas entre afirmativas e negativas, com respostas em escala Likert [Likert 1932], para averiguar o nível de usabilidade de um sistema. Pode ser usado em conjunto com outras escalas a fim de melhor avaliar um produto. Utiliza uma escala de conceitos e adjetivos para uma pontuação obtida a partir das respostas do questionário que vai de “pior imaginável” até “melhor imaginável”; e sua pontuação pode variar entre 0 e 100 (Figura 1).

Tabela 1. Taxonomia dos modelos de usabilidade. Fonte: [Falcão e Soares 2013]

Modelo	Dimensões	Definição	
Modelo de Eason (1984)	Tarefa	Frequência	Número de vezes que uma tarefa é realizada pelo usuário.
		Abertura	Extensão em que uma tarefa é modificável.
	Usuário	Conhecimento	O conhecimento que o usuário aplica na tarefa. Este deve ser apropriado ou não.
		Motivação	Como determinou ao usuário completar a tarefa.
	Sistema	Critério	A habilidade do usuário em não escolher usar alguma parte do sistema.
		Fácil de aprender	O esforço requerido para entender e operar um sistema não familiar.
Modelo de Shackel (1991)	Facilidade de aprendizado	Fácil de usar	O esforço requerido para operar um sistema uma vez que é entendido e dominado pelo usuário.
		Correspondência da tarefa	A extensão que cada informação e funções que um sistema fornece correspondem às necessidades do usuário para uma determinada tarefa.
	Eficácia	É descrita pelo intervalo da tarefa que deve ser melhor do que o nível de desempenho exigido, assim como uma porcentagem específica de um intervalo de usuários alvo dentro de uma gama específica do ambiente de uso.	
	Atitude	Corresponde ao tempo entre o treinamento dos usuários e suporte da instalação do sistema, incluindo o tempo de reaprendizado.	
Modelo de Nielsen (1993)	Flexibilidade	Corresponde a permissão de alguma variação de percentual especificado na tarefa e/ou no ambiente além do que foi especificado primeiramente.	
	Poucos erros	Corresponde aos níveis aceitáveis de custo humano em termos de cansaço, desconforto, frustração e esforço pessoal.	
	Facilidade de aprender	o sistema deve ser fácil de aprender para que o usuário, mesmo não tendo experiência, possa rapidamente começar a obter resultados satisfatórios do trabalho realizado.	
	Eficiência	está diretamente relacionada com a produtividade do sistema, de modo que uma vez que o usuário tenha aprendido o sistema, seja possível uma alta produtividade.	
	Satisfação	o sistema deve ser fácil de lembrar, de forma que o usuário ocasional não tenha que aprender tudo de novo sobre o sistema após algum período sem tê-lo usado.	
		o sistema deve ter uma baixa taxa de erros, de modo que os usuários cometam poucos erros durante o uso do sistema, e assim que erros sejam cometidos, estes possam ser corrigidos de forma simples e rápida. Além disso, erros catastróficos não devem ocorrer.	
		o sistema deve permitir uma interação agradável, para que os usuários estejam subjetivamente satisfeitos ao utilizá-lo.	

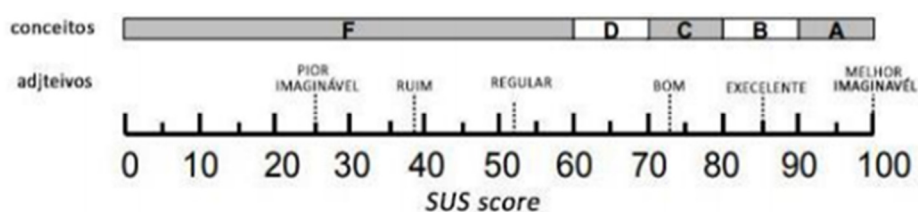


Figura 1. Escala de avaliação de usabilidade por SUS. Fonte: [Bangor, Kortum e Miller 2009]

Diferentemente de [Bangor, Kortum e Miller 2009], [Sauro 2011] analisou 500 estudos que utilizaram esta escala e aponta que a pontuação para um *software* ser considerado bom é 68 pontos. SUS já foi testado em *hardware*, *software*, *sites*, celulares, jogos e outros, logo se tornou um padrão da indústria com referências em diversas publicações [Sauro 2009]. A Figura 2 exemplifica questionamento de SUS.

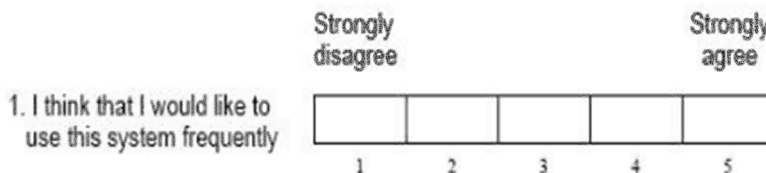


Figura 2. Exemplo de questão do SUS. Fonte: [Brooke 1996]

## 2.4. Trabalhos Relacionados

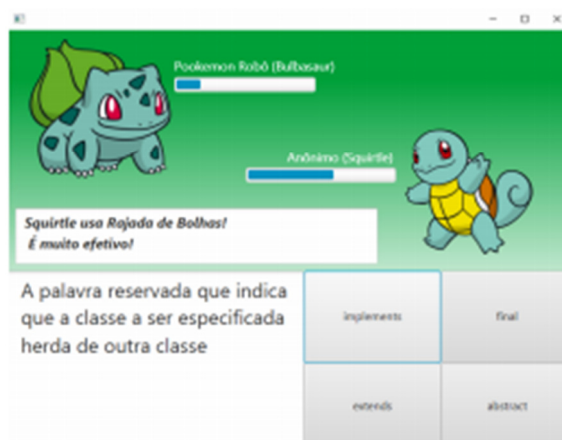
Vários jogos têm sido propostos na literatura com o intuito de auxiliar o processo de ensino e aprendizagem, sendo observadas diversas pesquisas relacionadas para auxiliar o ensino de Programação, desde conteúdos de disciplinas introdutórias até as mais avançadas, independentemente do paradigma de Programação.

Para a pesquisa de trabalhos presentes da literatura, utilizou-se a ferramenta *Google Scholar* através de *strings* de busca que continham palavras-chaves para: *softwares* educacionais, jogos educativos/educacionais, Programação, Orientação a Objetos, avaliação, usabilidade e *quiz*. A seguir, apresentam-se os trabalhos mais relevantes para esta pesquisa.

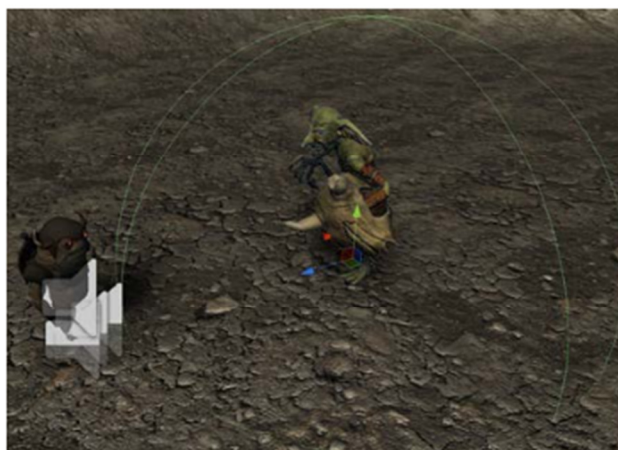
O Pookémon [Mombach et al. 2018] foi criado como um jogo *desktop* de perguntas e respostas sobre POO baseados nos jogos de Pokémon, porém a habilidade de vencer do jogador depende do seu conhecimento acerca do conteúdo, como observa-se na Figura 3. O jogo têm como objetivos: (i) possibilitar aos alunos aplicar os conhecimentos adquiridos; (ii) avaliar a usabilidade pedagógica e de *design* através de testes realizados com jogadores. Este jogo foi desenvolvido em sala de aula utilizando a linguagem Java. As ferramentas *Gluon Scene Builder* e *JavaFX* foram empregadas na construção da interface gráfica e planilha em formato *Comma-Separated Values* (CSV) armazena as questões. O jogo foi elaborado seguindo as fases de desenvolvimento: conceituação, produção e testes. A avaliação obteve resultados positivos, pois os jogadores acharam o jogo interessante e o *design* é coerente e atraente. Porém, os jogadores sugeriram melhorias, como a implementação de recompensas, trilha sonora e a inclusão de questões. Os desenvolvedores esperam disponibilizá-lo *on-line* para que os professores possam utilizar nas salas de aula.

Outro jogo presente na literatura, com intuito de auxiliar o ensino de POO, é o POOGame [Silva et al. 2016]. O jogo tem como objetivo ensinar os conceitos básicos de POO e foi modelado com base em outros jogos do tipo *Role Playing Game* (RPG), em que o jogador controla um personagem e, a partir de batalhas (Figura 4), é possível invocar criaturas e controlar suas ações utilizando comandos baseados na linguagem Java. O jogo foi desenvolvido através da *engine Unity3D*, utilizou-se a ferramenta Blender para os elementos da interface e do terreno do mundo virtual, e GitHub para realizar o controle de modificações. Os resultados apresentados demonstraram que o uso do jogo foi bem recebido pelos alunos e que ajudou a motivar a execução da fixação de

conceitos. [Silva et al. 2016] pretendem evoluir o jogo para englobar um maior número de conceitos de Orientação a Objetos (OO), e corrigir os *bugs* encontrados nos testes.



**Figura 3. Tela de Batalha do Jogo Pookémon.** Fonte: [Mombach et al. 2018]



**Figura 4. Campo de Batalha do Jogo POOGame.** Fonte: [Silva et al. 2016]

Além desses, também se identificou uma plataforma gamificada *web* e *mobile* de ensino de Lógica de Programação em desenvolvimento. O Play(code) [Barbosa et al. 2018] é uma plataforma que busca oferecer um ambiente de ensino que estimula o aprendizado, de Lógica de Programação, autodidata ou guiado por um tutor, utilizando perguntas e respostas, conforme observado na Figura 5. O desenvolvimento de Play(code) foi dividido em quatro fases: (i) a primeira fase foi o levantamento de literaturas que visavam estimular e motivar a utilização contínua de sistemas interativos; (ii) a segunda é relacionada à utilização de estratégias encontradas nas literaturas para serem utilizadas no contexto de sistemas educacionais, sendo essas estratégias no âmbito da colaboração, gamificação e *user experience* (UX), a serem adotadas no modelo de interface e interação do Play(code); (iii) a terceira fase foi a de levantamento dos requisitos funcionais e não funcionais; e, por fim, (iv) a quarta fase, em andamento, consiste no desenvolvimento do Play(code), em que a cada nova versão é avaliada por testadores e especialistas em Interação Humano Computador (IHC). Os desenvolvedores esperam que essa ferramenta possa auxiliar os alunos no aprendizado de Lógica de Programação, além de estimular diversas pessoas a buscarem conhecimento nessa área e contribuir tanto para o modelo de Educação à Distância, quanto para o presencial. Assim, os programadores desejam finalizar o Play(code) além

de avaliar e discutir o impacto do sistema como um recurso para auxiliar e estimular o aprendizado de Lógica de Programação.

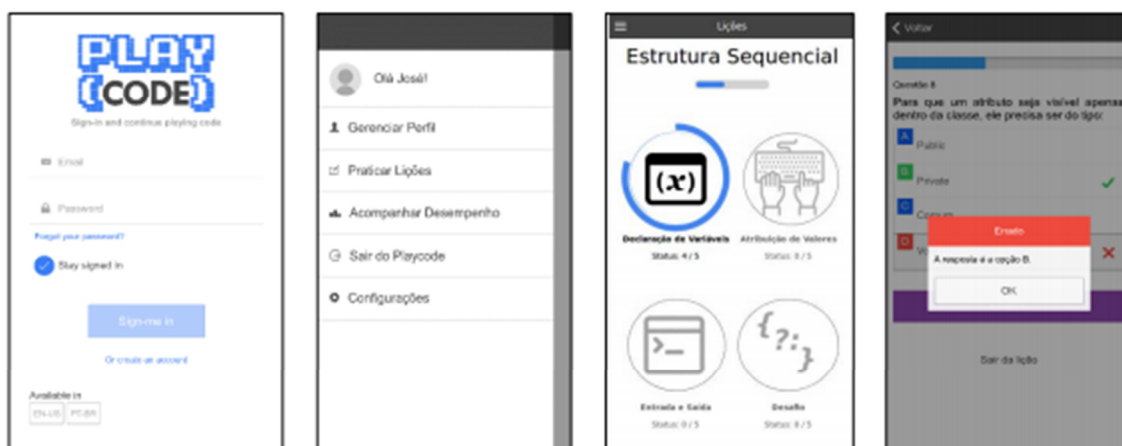


Figura 5. Exemplos de interface do Play(code). Fonte: [Barbosa et al. 2018]

#### 2.4.1. Análise Comparativa

Na Tabela 2 apresenta-se uma análise comparativa entre os trabalhos relacionados e o jogo desta pesquisa.

Tabela 2. Análise comparativa dos trabalhos relacionados

Autores	Ferramentas	Plataformas	Assuntos Abordados	Avaliação	Principais Resultados
[Mombach et al. 2018]	Gluon Scene Builder; JavaFX; Planilhas CSV	Desktop	Programação Orientada a Objetos	Avaliação qualitativa. Avalia a usabilidade pedagógica e de design.	A avaliação obteve resultados positivos, porém os jogadores sugeriram melhorias.
[Silva et al. 2016]	Unity3D; Blender; GitHub	Desktop	Programação Orientada a Objetos	Avaliação qualitativa. Avalia a motivação do jogo e a experiência do usuário.	O uso do jogo foi bem recebido; o jogo ajudou a motivar a execução da fixação de conceitos.
[Barbosa et al. 2018]	Não informado	Web; Mobile	Lógica de Programação	A plataforma segue o desenvolvimento em módulos e a cada nova versão é verificada por testadores e avaliada por especialistas em IHC.	Os desenvolvedores esperam que a ferramenta auxilie alunos no aprendizado de Lógica de Programação, além de estimular a busca de conhecimento nessa área e contribuir para o modelo de educação à distância e presencial.
Esta pesquisa	Java; Sprites; MVC; DAO XML.	Desktop	Programação Orientada a Objetos	Avaliação de usabilidade pedagógica e de design. Análise de confiabilidade dos dados. Teste de hipóteses. Análise de dependência.	O jogo obteve avaliação positiva nos diversos métodos aplicados. Os dados da pesquisa são confiáveis, e a avaliação independe da caracterização do indivíduo.

JAvatar possui funções como *quiz* e batalha, porém essas funcionalidades são diferentes dos trabalhos relacionados, como, por exemplo: (i) uso de gamificação, as perguntas geram pontos que podem ser utilizados para comprar cartas de defesa e/ou ataque, ou aumentar o poder dessas cartas; a; (ii) têm-se a utilização de avatares interativos, assim o jogador poderá visualizar quando está sendo atacado através de animações dos personagens (*sprites*); e (iii) pode-se gerenciar as perguntas utilizadas no jogo, as quais estão dispostas em arquivos XML.

### 3. Métodos

#### 3.1. O Jogo JAvatar

##### 3.1.1. Conceito

JAvatar originou-se de uma ideia de disciplina (Interface Homem-Máquina), como um jogo de perguntas e respostas, em que as ações dos personagens dependeriam das respostas do quiz do jogador. Utiliza a fantasia de jogos de batalha como Street Fighter, em que personagens disputam partidas. Cada jogador tem a possibilidade de atacar o oponente ou se defender de um ataque. À partida é finalizada quando a vida de um dos personagens for zerada. É possível customizar um avatar e um nome para o jogador.

A primeira versão [Veras et al. 2020] foi avaliada em um experimento controlado, com 20 alunos de Bacharelado em Sistemas de Informação, os quais já haviam estudado POO. Realizou-se uma avaliação quali-quantitativa com coleta de dados por questionário e entrevista.

Através do teste de Qui-Quadrado, obteve-se um p-valor de 0,0234 para 4 graus de liberdade e nível de significância de 5%, corroborando a hipótese de o jogo contribuir para o processo de ensino e aprendizagem de POO, uma vez que há dependência entre usabilidades pedagógicas e design com níveis de aceitação, sendo este último positiva em sua grande maioria. Também foi classificado como um *software* de qualidade “excelente” por SUS, com uma pontuação média de 78,25.

Em busca de tornar o jogo mais dinâmico, especialistas em IHC sugeriam modificar o jogo para melhorar a gamificação, adicionando ao jogo de batalha a utilização de cartas, como o jogo Yu-Gi-Oh!. Essas estão relacionadas à possibilidade de atacar e defender conforme a carta escolhida, e para utilização dessas cartas o jogador deveria acumular pontos obtidos a partir de acertos das perguntas do *quiz*.

##### 3.1.2. Desenvolvimento

Para a versão 2.0 de JAvatar, utilizou-se o desenvolvido iterativo e incremental, com prototipação a partir de cenários por *storyboards*, visando atender as recomendações qualitativas da primeira versão. A nova versão é desenvolvida com codificação através da Linguagem de Programação Java [Oracle 2012] e utiliza boas práticas como a organização a partir do padrão arquitetural de *software Model-View-Controller* (MVC) [Burbeck 1992], para facilitar a manutenção do sistema; e *Data Access Object* (DAO) para persistência de dados. O gerenciamento e armazenamento de conteúdos relativos às perguntas e respostas do quiz são dados em arquivos *Extensible Markup Language* (XML) [Quin et al. 2012], permitindo adicionar, remover e atualizar perguntas sem a



necessidade de interfaces complexas e ou utilização de sistema gerenciados de banco de dados.

Escolheu-se a abordagem de [Sakamoto e Nakajima 2013] que, segundo o autor, a utilização de personagens de animações conhecidos, atrai e motiva os participantes para jogar. Quando os jogadores escolhem seus personagens favoritos, por algum vínculo emocional, não querem que o seu personagem perca, os motivando a ganhar e finalizar o jogo.

Foram utilizadas soluções de animação pela técnica de jogos digitais conhecida como *sprites*, além da ilusão de um cenário 3D através da organização em camadas e sobreposição de imagens 2D, aumentando assim a interação com o jogo.

### 3.1.3. Regras

As regras de JAvatar v. 2.0 basearam-se em jogos de perguntas e respostas; e jogos de cartas como YU-Gi-Oh!. O jogo é *singleplayer* em que o jogador batalha uma partida (*round*) em modo “*arcade*”, contra o computador. Cada ação é dada em um turno que, após responder uma pergunta do quiz, poderá promover um ataque contra o adversário, defender-se do ataque do rival ou apenas acumular pontos.

O jogador tem direito a responder uma pergunta por turno ou pulá-la, porém só poderá pular até três questões. Caso a resposta escolhida esteja correta, o jogador acumulará 10 pontos à sua pontuação, a qual poderá ser utilizada para comprar cartas de ataque, defesa ou magia, sendo esta última uma possibilidade de aumentar o poder das anteriores. Cada jogador terá 5 cartas em seu deck que deverão ser usadas a partir da estratégia do jogador. Caso não se tenha pontuação suficiente para comprar uma carta, o jogador deverá finalizar seu turno e esperar a ação do adversário. A partida finaliza quando a vida de um dos personagens é zerada após sucessivos ataques sem defesas.

## 3.2. Avaliação de JAvatar

Para realizar os procedimentos de avaliação de JAvatar, foi promovida uma pesquisa experimental transversal de avaliação de *software* com a coleta de dados através da aplicação de questionário disposto em formulário<sup>1</sup> *on-line* a indivíduos de cursos de Computação para um estudo quali-quantitativo através da plataforma *Google Forms*. A escolha por uma abordagem remota está atrelada ao fato da impossibilidade de um experimento controlado, como realizado na avaliação da primeira versão do jogo, em virtude da pandemia da COVID-19.

A distribuição do formulário foi dada por convite enviado a e-mails de grupos e mídias sociais, assumindo-se uma amostragem não-probabilística e por conveniência, por ser uma técnica de fácil operacionalização e baixo custo, mas acredita-se que por ser tratar de público específico não hajam imposições de diferentes estratos que comprometam o estudo, ou seja, não introduzirá vieses em relação à população.

O questionário (Tabelas 3, 4 e 5) contém um total de 29 questionamentos divididos em três grupos, a saber: as questões de 1 a 9 destinam-se a traçar o perfil dos usuários; as questões de 10 a 24 são sobre análise design do jogo; e as questões de 25 a 29 são sobre usabilidade pedagógica.

---

<sup>1</sup> <https://forms.gle/FSKJCJ6Me5dcoHKS56>

**Tabela 3. Questionário de identificação de perfil de usuário**

Nº	Questões
Q1	Qual sua idade?
Q2	Qual seu sexo?
Q3	Qual é seu perfil?
Q4	Qual sua área de atuação?
Q5	Qual o seu grau de instrução?
Q6	Você já estudou disciplinas com conteúdos relacionados à Programação Orientada a Objetos (POO)?
Q7	Qual seu nível de conhecimento em POO?
Q8	Quais dispositivos você costuma utilizar para estudar POO?
Q9	Com que frequência você costuma utilizar softwares educacionais para auxiliar nos estudos de POO?

**Tabela 4. Questionário de usabilidade de design**

Nº	Questões	Heurísticas
Q10	Foi fácil aprender a usar o jogo.	Facilidade de aprendizado
Q11	O jogo permite que usuários sem experiência sejam capazes de explorá-lo e aprender rapidamente como obter resultados com ele.	
Q12	É fornecido <i>feedback</i> adequado e em tempo razoável sobre a ação do usuário.	Visibilidade de status do sistema
Q13	Consegue-se compreender, de forma clara, o funcionamento do jogo.	
Q14	Os significados de símbolos e ícones são compreensíveis e intuitivos.	Correspondência entre o Sistema e o Mundo Real
Q15	As informações do jogo estão dispostas em uma ordem lógica e natural.	
Q16	Os formatos de apresentação das opções de jogo são mantidos consistentes entre os jogadores.	Consistência e Padrões
Q17	O sistema possui padrões e estilos consistentes.	
Q18	O jogo utiliza textos e ícones em linguagem habitual e conhecida em jogos digitais.	Reconhecer ao invés de lembrança
Q19	As ações do personagem fazem sentido para você.	
Q20	O jogo exibe quantidade necessária de informação e opção em cada tela, sem texto ou imagens em excesso.	Estética e <i>Design</i> Minimalista
Q21	As telas são simples e objetivas.	
Q22	O jogo possui opção de ajuda adequada ao seu entendimento.	Ajuda e Documentação
Q23	O jogo possui elementos de ajuda e documentação ao usuário.	

**Tabela 5. Questionário de usabilidade pedagógica**

Nº	Questões
Q24	O jogo apresenta desafios pedagógicos adequados ao aprendizado de POO, testando diferentes níveis de conhecimento do usuário.
Q25	A ambientação do jogo (Batalha de Personagens) é atrativa e desperta no usuário interesse pelo conteúdo abordado.
Q26	O jogo é atrativo, envolve e motiva ao desafio de concluir o jogo.
Q27	O uso do jogo contribui para o aprendizado
Q28	A utilização do jogo é estimulante quando comparada a outras atividades tradicionais da disciplina.
Q29	Eu jogaria o jogo novamente.

Antes de responder as questões (10 a 24), o usuário realiza tarefas definidas no formulário e a partir desta experiência responder aos questionamentos. Ao término da avaliação, tem-se a possibilidade dos indivíduos expressarem suas opiniões e sugestões sobre o jogo em campo específico, bem como da atribuição de uma nota para o jogo a partir da experiência vivenciada, sendo esta nota chamada de Experiência Geral.

A partir das questões do primeiro grupo será possível realizar a caracterização dos indivíduos e verificar a possibilidade de fazer parte da avaliação do *software*, uma vez que se assume, especialmente, a premissa que esses devem ter estudado disciplinas relacionadas à POO, caso contrário, as respostas serão desconsideradas na análise.

As questões para o critério de usabilidade de princípios de *design* são dadas a partir das heurísticas de Nielsen [Nielsen 1993; Nielsen 1994]. Entretanto, nem todas as heurísticas foram utilizadas para a avaliação, uma vez que não são contempladas no projeto de interface.

As questões da avaliação são baseadas e adequadas a partir da literatura pesquisada [Breyer 2008; Ghelardi et al. 2012; ISO 2001; Lopes 2012; Medeiros 2014; Mombach et al. 2018; Oliveira e Savoie 2011; Raposo 2018; Valle et al. 2013], pois se tem o entendimento que essas já são previamente validadas, não necessitando da realização de um pré-teste.

As questões 10 a 29 foram escritas em modo afirmativo, com a disposição de respostas em escala Likert [Likert 1932] como métrica de avaliação, na qual um usuário participante da pesquisa responderá um questionamento em uma escala de 1 a 5, variando na concepção de: 1 - Discordo totalmente (DT), 2 - Discordo (D), 3 - Neutro (N), 4 - Concordo (C) e 5 - Concordo totalmente (CT).

Antes de realizar a análise de usabilidade de JAvatar, primeiramente, verifica-se a confiabilidade dos dados, ou seja, a qualidade das respostas do questionário. Para isso será utilizado o coeficiente ( $\alpha$ ) de Alfa de Cronbach [Cronbach 1951, George e Mallery 2003, p. 231], que segundo [George e Mallery 2003, p. 231], o valor de  $\alpha$  indica a qualidade dos dados, podendo variar entre 0 e 1 e tem-se a seguinte classificação para os valores de  $\alpha$ : se  $\alpha > 0.9$  – Excelente;  $\alpha > 0.8$  – Bom;  $\alpha > 0.7$  – Aceitável;  $\alpha > 0.6$  – Questionável;  $\alpha > 0.5$  – Ruim; e  $\alpha < 0.5$  – Inaceitável.

Para avaliar a usabilidade de JAvatar serão utilizadas duas abordagens: qualidade do *software* por SUS e inferência estatística por testes de hipóteses.

É método de SUS [Brooke 1996]: (i) nas questões ímpares a pontuação da resposta da pergunta corresponde a posição na escala menos 1; (ii) nas pares a pontuação corresponde a posição na escala subtraída de 5; (iii) em seguida, para cada usuário participante soma-se as pontuações de cada questão e depois multiplica-a por 2,5; e, por fim, (iv) obtêm-se a média das pontuações de todos os participantes, sendo esta a pontuação final.

Para SUS é necessário redefinir o fator multiplicativo estabelecido por [Brooke, 1996], que é de 2,5 para a obtenção da pontuação máxima de 100 pontos. Segundo o autor, para as 10 questões, assumindo que todas as afirmações sejam positivas (5 para o nível de concordância total) para obtenção da pontuação máxima:  $4 \times 10 = 40$ , e assumindo o fator multiplicativo de 2,5, tem-se uma pontuação de 100 pontos ( $40 \times 2,5 = 100$ ). Entretanto, como o questionário desta pesquisa para usabilidade de design possui 14 questões (Tabela 4), na avaliação de JAvatar, o fator multiplicativo será de 1,7857, pois:  $14 \times 4 = 56$  e  $100 / 56 \cong 1,7857$ .

Ressalta-se o fato de que SUS utiliza as perguntas ímpares na forma afirmativa e os pares na negativa. Porém, como as questões deste trabalho foram dispostas de maneira afirmativo-positiva, então foi necessário inverter o valor das respostas das

perguntas pares, ou seja, as respostas com valor 1 (discordo totalmente) se transformaria em 5 (concordo totalmente), 2 (discordo) em 4 (concordo), e assim sucessivamente.

Em testes de hipóteses é necessário fundamentar as hipóteses. Para isso, definem-se três situações. Na Situação 1 (S1) será analisada se há diferença significativa entre as médias das avaliações de percepção de Experiência Geral em relação a SUS. São as hipóteses: (i) hipótese nula ( $H_{0-S1}$ ): as médias das notas dadas pelos usuários para Experiência Geral e SUS são consideradas estatisticamente iguais; e (ii) hipótese alternativa ( $H_{1-S1}$ ): as médias das notas dadas pelos usuários para Experiência Geral e SUS diferem estatisticamente. Para responder à Situação 1, tem-se a Análise de Variância (ANOVA) a partir do Teste F para verificação das hipóteses [Vieira 2006].

Na Situação 2 (S2) será verificada a existência de dependência entre os aspectos definidos no questionário e o nível de aceitação. Havendo essa dependência e se observado uma alta positividade em relação às afirmações, pode-se então verificar as seguintes hipóteses. (i) hipótese nula ( $H_{0-S2}$ ): a usabilidade oferecida pelo JAvatar não fornece contribuição significativa para o processo de ensino e aprendizagem de POO; e (ii) hipótese alternativa ( $H_{1-S2}$ ): a usabilidade oferecida pelo JAvatar fornece contribuição significativa para o processo de ensino e aprendizagem de POO.

Visando analisar se há relação de dependência de variáveis de caracterização de indivíduo em relação à avaliação de JAvatar, pretende-se verificar, Situação 3 (S3), as seguintes hipóteses: (i) hipótese nula ( $H_{0-S3}$ ): avaliação de JAvatar é independente de variável de caracterização é indivíduo; e (ii) hipótese alternativa ( $H_{1-S3}$ ): avaliação de JAvatar é dependente de variável de caracterização é indivíduo. As implicações de dependência são dadas as variáveis de caracterização: idade, sexo, perfil do usuário, área de atuação, grau de instrução, nível de conhecimento em POO, dispositivo comumente utilizado para estudar POO, uso de *softwares* educacionais para auxiliar nos estudos de POO. Essas variáveis são correspondentes às questões do perfil do usuário.

Para as Situações 2 e 3 será utilizado o teste hipóteses por Qui-Quadrado ( $\chi^2$ ) [Plackett 1983 apud Levin e Fox 2004]. Vale ressaltar que se em alguma tabela de contingência algum valor das células refletirem valores inferiores a 5 na frequência esperada e, conforme [Siegel 1975, p. 124], em tabelas  $rxk$ , (várias linhas x várias colunas), o teste Qui-quadrado pode ser aplicado somente se o número de células com frequência esperadas inferior a 5 é 20% do total das células e se nenhuma célula tem frequência esperada inferior a 1. Se essas condições não são satisfeitas pelos dados na forma em que foram coletados originalmente, o pesquisador deve combinar categorias adjacentes de modo a aumentar as frequências esperadas nas diversas células.

O detalhamento de todos os métodos e resultados obtidos da pesquisa é encontrado em [Veras e D'Emery 2021].

## **4. Resultados e Discussões**

### **4.1. Elementos Visuais e Interativos de JAvatar**

A ilustração da segunda versão de JAvatar é apresentada nas Figuras 6, 7, 8 e 9.

A tela inicial (Figura 6a) é composta pelos botões de jogar, customizar, som, ajuda, tutorial online, informações do jogo e sair. Ao clicar em customizar o jogador

deve escolher o avatar de seu personagem e inserir o *nick* (Figura 6b); e ao clicar em jogar (Figura 6c) o adversário é sorteado pelo sistema e a batalha inicia em 5 segundos.

Na tela do tutorial, Figura 7, o jogador poderá ler o tutorial do jogo. Há os botões de avançar e voltar, para que os usuários possam ler todas às instruções do tutorial; o botão home volta à tela de abertura; o botão pular para não ver as instruções; e jogar para iniciar uma batalha. Esta tela só é exibida na primeira utilização, porém poderá ser acessada na tela de abertura ou na tela de batalha (Figura 8).

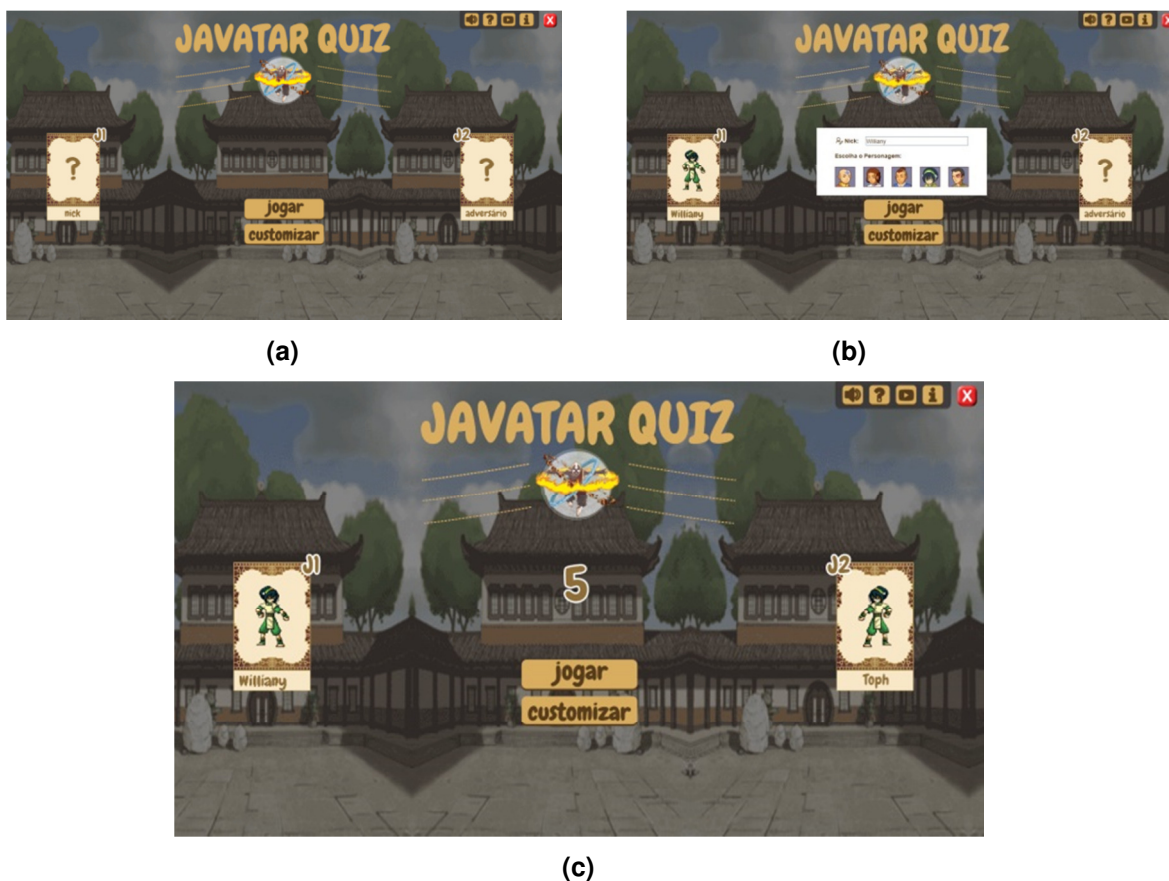


Figura 6. Tela Inicial de JAvatar: (a) abertura; (b) personalização de jogador; e (c) personagens de batalha.



Figura 7. Tela do Tutorial

Na tela de batalha, Figura 8a, é possível observar o módulo *quiz*, em que o jogador deverá responder aos questionamentos sobre POO. Ao acertar ou errar uma pergunta, a tela é atualizada (Figura 8b) com um ícone de confirmação de acerto ou erro e modificação da pontuação. A Figura 8c ilustra a possibilidade de o jogador escolher cartas de defesa, ataque ou magia, enquanto a Figura 8d ilustra as cartas compradas pelo o jogador para utilizá-la em seu turno.

Ao final do jogo é mostrada uma mensagem de vitória ou derrota (Figura 9). É possível duelar novamente contra o mesmo jogador clicando no ícone jogar novamente ou voltar à tela de abertura e clicar em jogar, mas o sistema escolherá outro adversário.

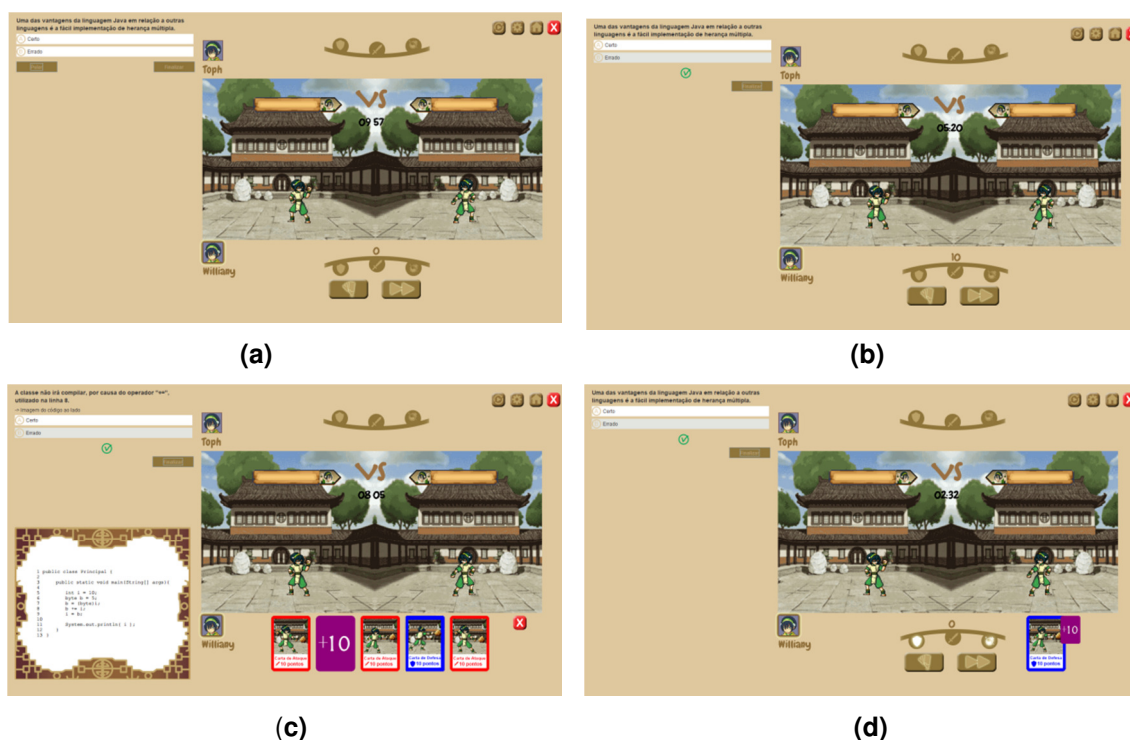


Figura 8. Telas de JAvatar: (a) módulo quiz; (b) feedback de resposta e pontuação; (c) deck com cartas para comprar; e (d) carta de ação ativada.



Figura 9. Telas de JAvatar de confirmação de: (a) vitória; e (b) derrota.

## 4.2. Resultado da Avaliação

Nesta seção, apresentam-se os resultados obtidos a partir das respostas dos participantes para o questionário, em especial, sobre o perfil e concordâncias.

No total, participaram 25 indivíduos na avaliação do *software* JAvatar, entretanto 24 foram considerados, pois 1 afirmou não ter estudado POO. A amostra é composta por: (i) 25% de mulheres e 75% de homens; (ii) a maioria dos participantes são alunos, 79%; (iii) sendo maioria na área de Computação (95,8%); (iv) já a maioria dos participantes tem entre 21 e 24 anos (58,3%), as idades inferiores a 21 anos é de 12,6% e as superiores a 24 é de 29,3%; (v) a maioria dos participantes possuem o superior incompleto (62,5%), ou seja, que ainda estão se graduando em algum curso superior. Vale destacar a existência de participantes do ensino médio (12,7%), uma vez que se trata de alunos em que tem em seus cursos disciplinas de Programação, em especial, Orientada a Objetos; (vi) apesar de todos os participantes afirmarem já ter visto POO, o grau de conhecimento afirmado pelos participantes é de 33,4% para graus 2 e 3, e de 66,6% para 4 e 5; (vii) todos os participantes utilizam computadores pessoais para estudar POO e desses, apenas 25% utilizam dispositivos móveis para estudar; e (viii) apesar do uso de tecnologias para o estudo, 54,2% dos participante aponta que raramente utilizam jogos educacionais para estudos de POO.

### 4.3. Análise e Interpretação da Avaliação de JAvatar

Antes de iniciar a avaliação de JAvatar, conforme definido na Seção 3, as respostas dos usuários serão verificadas quanto à sua qualidade através do coeficiente alfa de Cronbach ( $\alpha$ ) [Cronbach 1951, George e Mallery 2003, p. 231]. A Tabela 6 apresenta os coeficientes obtidos.

**Tabela 6. Alfa de Cronbach para as respostas dos usuários**

Usabilidade	Nº de Questões (n)	Nº de Participantes	Total de respostas	$\alpha$	Qualidade
Princípios de design	14	24	336	0,903	Excelente
Pedagógica	6	24	144	0,723	Aceitável
Todas as Questões	20	24	480	0,919	Excelente

Portanto, afirma-se que os dados levantados por meio do questionário (Tabelas 4 e 5), são tidos como um instrumento de medida confiável, uma vez que possui uma confiabilidade alta.

Para as análises, a Tabela 7 apresenta as frequências observadas, ou seja, as respostas dos questionários já se considerando a inversão dos valores das respostas das perguntas pares na usabilidade de design, conforme definido na Seção 3.

**Tabela 7. Frequências observadas**

Métricas	Usabilidade		Total
	Pedagógica	Design	
Discordo totalmente	0	123	123
Discordo	0	38	38
Neutro	5	18	23
Concordo	25	41	66
Concordo totalmente	114	116	230
<b>Total</b>	<b>144</b>	<b>336</b>	<b>480</b>

A pontuação SUS alcançada por JAvatar foi de 77,67, o que classifica, portanto, sua usabilidade de design como “excelente”. Para a percepção de Experiência Geral dos usuários, a média das notas atribuídas foi de 88,75, ambas apresentadas na Tabela 8.

**Tabela 8. Comparativa das pontuações de Experiência Geral e SUS**

Avaliação	Pontuação
SUS	77,67
Experiência Geral	88,75

Apesar das diferenças observadas na Tabela 8, para verificação das hipóteses da Situação 1, ou seja, sobre a avaliação de usabilidade com ênfase na Experiência Geral e SUS por ANOVA a partir do Teste F, verificou-se a rejeição de  $H_{0-S1}$ . O valor-P encontrado é próximo à zero (5,88E-05), portanto inferior ao nível de significância de 5% ( $p < 0,05$ ), então se confirma  $H_{1-S1}$ , ou seja, as médias obtidas das avaliações entre os grupos diferem estatisticamente. Logo as avaliações são dependentes da experiência do utilizador.

Para a análise de inferência estatística da Situação 2, obteve-se  $\chi^2_{\text{calculado}} = 113,624$ . Como  $\chi^2_{\text{tabelado}} = 9,488$  (com 4 graus de liberdade e nível de significância definida de 5%), então se refuta  $H_{0-S2}$ , pois  $\chi^2_{\text{calculado}} > \chi^2_{\text{tabelado}}$ . Portanto, a usabilidade oferecida por JAvatar fornece contribuição significativa para o processo de ensino e aprendizagem de POO ( $H_{1-S2}$ ), tendo em vista a forte dependência ( $p < 0,01$ ) entre as variáveis de usabilidade e o grau de aceitação, além do fato do nível de aceitação corroborar a qualidade dos questionamentos definidos para usabilidade.

Para verificação da Situação 3, foi necessário realizar a combinação de categorias adjacentes de modo a aumentar as frequências esperadas nas diversas células, conforme [Siegel 1975, p. 124] e apresentado na Seção 3. Sendo assim destacam-se as tabelas de contingência com categorias combinadas a partir das variáveis: (i) idade possui 2 categorias (idade < 25 e idade  $\geq$  25) e 5 níveis de concordância; (ii) perfil possui duas categorias (graduando e graduado) e 3 níveis de concordância; (iii) área de atuação possui duas categorias (área de Computação e outras áreas) e 3 níveis de concordância; (iv) grau de instrução possui 2 categorias (superior incompleto/médio e superior completo) e 3 níveis de concordância; (v) nível de conhecimento do usuário possui 2 categorias (2/3 e 4/5 – em que 1 pouco e 5 é muito) e 5 níveis de concordância; (vi) dispositivos comumente usados para estudar POO possui duas categoria (apenas computador e computador e dispositivo mobile) e 5 níveis de concordância; (vii) uso de *softwares* educacionais para estudar POO possui três categorias (nunca, raramente, às vezes/frequentemente) e 5 níveis de concordância).

Quanto aos 3 níveis de concordância definidos na combinação de categorias, têm-se: discordância (combinação de discordo totalmente e discordo), neutro e concordância (combinação de concordo totalmente e concordo).

Na Tabela 9 é apresentado um resumo das análises de dependência das variáveis. A marcação nas células com “Dependente” configura que foi identificado uma dependência entre a variável e a usabilidade pedagógica e de design, ou seja, no teste Qui-Quadrado o valor calculado de p é inferior a 0,05 ( $p < 0,05$ ).



**Tabela 9. Resumo do efeito das variáveis de caracterização de indivíduo sobre as de usabilidade**

Variável	Usabilidade	
	<i>p</i> (teste $\chi^2$ )	Análise
Idade	0,0469	Dependente
Sexo	0,9933	Não dependente
Perfil do usuário	0,1366	Não dependente
Área de atuação	0,5913	Não dependente
Grau de instrução	0,6756	Não dependente
Nível de conhecimento em POO	0,2749	Não dependente
Dispositivos comumente usados para estudar POO	0,00034	Dependente
Uso de softwares educacionais para auxiliar nos estudos de POO	0,9867	Não dependente

Portanto, apenas as variáveis “idade” e “dispositivos comumente usados para estudar POO” em relação à usabilidade tiveram as hipóteses nulas ( $H_{0-S3}$ ) refutadas, apontando a dependência entre as variáveis.

## 5. Conclusão

Este estudo apresentou o desenvolvimento e a avaliação de usabilidade de um jogo educacional, e os autores acreditam que iniciativas, tais como essa, contribuam na criação de soluções que possam estimular o aprendizado de conteúdos em cursos de Computação.

Ressaltasse a importância dos resultados dessa pesquisa, pois pesquisadores no âmbito de IHC reforçam quão necessária é a utilização de estratégias de avaliação em diferentes tipos de *software*, inclusive os educacionais, para delinear suas vantagens e desvantagens do seu uso [Silva et al. 2015], principalmente, a partir de uma abordagem que apresente a utilização de técnicas quantitativas, comumente não observadas em apresentações de *software* no contexto educacional.

Ficou evidente a necessidade de se estimular alunos a conhecerem ferramentas que possam auxiliar nos estudos. Em se tratando de cursos de Computação, sabe-se das altas taxas de retenção e desistência, principalmente por envolverem conteúdos comumente não vistos na educação básica. Essa afirmação é corroborada pelas evidências apresentadas, ou seja, apesar dos indivíduos da avaliação utilizar recursos tecnológicos para o estudo, uma grande maioria não tem o hábito de utilizar jogos, dos quais são considerados recursos de alto poder motivacional.

Os aspectos de avaliação realizados apontam a qualidade do jogo JAvatar. Entretanto, os autores sabem que a utilização de amostragem não-probabilística e por conveniência pode ter como consequência a incapacidade de fazer afirmações gerais com rigor estatístico sobre a população, mas esse tipo abordagem foi necessária, uma vez que a pesquisa decorre-se em um período atípico que inviabiliza o contato direto com usuários, em decorrência do momento atual vivido pelo mundo, ou seja, a pandemia da COVID-19. Ainda assim, acredita-se que a abordagem utilizada é válida e não desqualifica a pesquisa realizada.

Para os trabalhos futuros esperar-se realizar melhorias em JAvatar, a partir de uma análise secundária das sugestões apresentadas pelos participantes da avaliação. Entretanto, a versão atual já pode ser utilizada como ferramenta para auxiliar no processo de ensino e aprendizagem de aulas que abordam POO.

## Referências

- ABNT. (2002). ABNT NBR ISO 9241-11:2011. Requisitos ergonômicos para trabalho de escritórios com computadores: orientações sobre usabilidade, Rio de Janeiro.
- Bangor, A., Kortum, P. T. and Miller, J. T. (2009). Determining what individual SUS scores mean: adding an adjective rating scale. In: *Journal of Usability Studies*, 4 ed, p.114-123.
- Barbosa, G. A. R., Silva, I. S., Coutinho, F. R. S., Froes, P., Barbosa, R., Freiras, F., Nascimento, V. M. P., Bulgarelli, M., Marques, G., Renan, J. P., Ribeiro, P. M., Sena, J. e Franco, D. (2018). “Play(code): Uma proposta para estimular o aprendizado de Lógica de Programação”, In: *Anais do XXIX Simpósio Brasileiro de Informática na Educação*, Fortaleza, p. 1776-1779.
- Breyer, F. B. (2008). “Avaliação Heurística para Protótipos de Jogos Digitais”, Dissertação (Mestrado) – Universidade Federal de Pernambuco, CAC, Design, 103 f.
- Brooke, J. (1996). SUS: A “quick and dirty” usability scale. In Jordan, P. W., Thomas, B., Weerdmeester, B. A. and McClelland, A. L. (Eds.), *Usability Evaluation in Industry*. London: Taylor and Francis.
- Burbeck, S. (1992). *Applications Programming in Smalltalk-80™: How to use Model-View-Controller (MVC)*. Softsmarts Inc.
- Clark, R. e Mayer, R. E. (2012). *E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning (3rd ed.)*. Pfeiffer & Company.
- Cronbach, J. L. (1951). Coefficient alpha and the internal structure of tests. In: *Psychometrika*, v. 16. n. 3, p. 297-334.
- Cybis, W., Betiol, A. H. e Faust, R. (2010). *Ergonomia e Usabilidade: conhecimentos, métodos e aplicações*. 2. Ed. São Paulo: Novatec Editora.
- Falcão, S. e Soares, M. M. (2013). Usabilidade de Produtos de Consumo: uma análise dos conceitos, métodos e aplicações. In: *Estudos em Design*. Rio de Janeiro: v. 21, n. 2, p. 01 – 26.
- Filho, A. M. S. (2008). Usabilidade de Software. In: *Engenharia de Software Magazine*, Brasil, 2008.
- George, D. e Mallery, P. (2003). *SPSS for Windows step by step: A simple guide and reference*. 11.0 update (4th ed.). Boston: Allyn & Bacon. In: Gliem, J. A., Gliem, R. R. (2003). Calculating, Interpreting, and Reporting Cronbach’s Alpha Reliability Coefficient for Likert-Type Scales, Midwest Research-to-Practice Conference in Adult, Continuing, and Community Education, The Ohio State University, Columbus, OH, p. 82-88.
- Ghelardi, A., Otsuka, J. L. e Kawakami, C. (2012). “Acessibilidade na Educação a Distância: desenvolvimento de um player de mídia acessível utilizando HTML5 e WAI-ARIA e sua integração com o Moodle”, In: *Anais do 23º Simpósio Brasileiro de Informática na Educação*, Rio de Janeiro, Novembro.

- Henrique, S. M. e Rebouças, A. D. S. (2015). Objetos de Aprendizagem para auxiliar o ensino de conceitos do Paradigma de Programação Orientada a Objetos. In: Revista Novas Tecnologias na educação, v. 13, n. 2, p. 1-10.
- ISO. (2001). ISO/IEC 9126-1:2001. Software engineering – Product quality – Part 1: Quality Model, <https://www.iso.org/standard/22749.html>.
- Levin, J. e Fox, J. A. (2004). “Estatística para Ciências Humanas”, São Paulo: Prentice Hall, 9 ed.
- Likert, R. (1932). A technique for the measurement of attitudes. In: Archives of Psychology. n. 140, p. 44-53.
- Lopes, A. C. B. (2012). “Desenvolvimento de um Jogo Didático para Ensino e Programação Orientada a Objetos e sua Aplicação em Cursos Técnicos de Computação”, Dissertação (Mestrado em Ciência da Computação) – Universidade Federal Rural do Semi-Árido, 24 f.
- Medeiros, T. (2014). Um Framework para criação de jogos voltados para o ensino de lógica de programação. Dissertação (Mestrado em Sistemas e Computação) - Centro de Ciências Exatas e da Terra, Universidade Federal do Rio Grande do Norte, Natal, 80 f.
- Mombach, J. G., Castro, B. M. N., Santos, E. S. e Santos, M. B. B. (2018). “POOkemon: um jogo sobre programação orientada a objetos”, In: Anais do XVII Simpósio Brasileiro de jogos e Entretenimento Digital (SBGames 2018), Foz do Iguaçu, Novembro.
- Morais, R. X. T. (2003). Software educacional: A importância de sua avaliação e de seu uso na sala de aula. Monografia (Bacharelado em Ciência da Computação) – Faculdade Lourenço Filho, 52 p.
- Nielsen, J. (1993). Usability Engineering. EUA: Morgan Kaufmann.
- Nielsen, J. (1994). “10 Usability Heuristics for User Interface Design”, <https://www.nngroup.com/articles/ten-usability-heuristics/>, Dezembro.
- Nielsen, J. e Molich, R. (1990). Heuristic evaluation of user interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Seattle, EUA, Abril, p. 249-256.
- Oliveira, H. e Savoie, M. (2011) “Aplicação do Método de Heurística no Sistema Colaborativo HEDS”, In: Revista Científica do ITPAC, v.4, n.3, P1, Araguaína - Brasil.
- Oracle. (2012). Java Programming Language, <http://docs.oracle.com/javase/7/docs/technotes/guides/language/index.html>
- Piteira, M. e Haddad, S. R. (2011). Innovate in your program computer class: An approach based on a serious game. In: Proceedings of the 2011 Workshop on Open Source and Design of Communication, OSDOC '11, New York, NY, USA, p 49–54.
- Plackett, R. L. (1983). Karl Pearson and the Chi-Squared Test. In: International Statistical Review. v. 51, n. 1, p. 59–72.
- Preece, J., Rogers, Y. e Sharp, H. (2007). “Design de interação: além da interação homem – computador”. Porto Alegre: Bookman.

- Quin, L. et al. (Ed.). (2007). XML Query (XQuery). W3C – World Wide Web Consortium, <http://www.w3.org/XML/Query/>
- Raposo, A. (2018). “Avaliação Heurística: Introdução à Interação Humano-Computador”, PUC, Rio de Janeiro.
- Rocha, H. V., e Baranauskas, M. C. C. (2003). Design e Avaliação de Interfaces Humano Computador. Unicamp, São Paulo.
- Sakamoto, M. e Nakajima, T. (2013). Augmenting Yu-Gi-Oh! Trading Card Game as Persuasive Transmedia Storytelling. In: Second International Conference, DUXU 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA.
- Sauro, J. (2009) “Measuring Usability With The System Usability Scale (SUS)”, <https://measuringu.com/sus/>, Setembro.
- Sauro, J. (2011). Does prior experience affect perceptions of usability?, <https://measuringu.com/prior-exposure/>
- Siegel, S. (1975). Estatística Não-paramétrica Para as Ciências do Comportamento. São Paulo: McGraw-Hill.
- Silva, J. C. R., Oliveira, E. R. e Barbosa, G. A. R. (2015). Caracterização de Estratégias de Gamificação em Aplicativos Móveis Educacionais: Um Estudo de Caso do Aplicativo Duolingo. In: Brazilian Symposium on Human Factors in Computing Systems, Salvador.
- Silva, L., Bonfim, B., Silva, R., Silva, J., Moura, W., Bezerra, B. e Jucá, P. (2016). “Poogame: Um jogo sério para o ensino de programação orientada a objetos”, In Anais do XXIV Workshop sobre Educação em Computação 2016, Porto Alegre, p. 338–347.
- Valle, P. H. D., Vilela, R. F., Júnior, P. A. P. e Inocêncio, A. C. G. (2013). HEDEG - Heurísticas para Avaliação de Jogos Educacionais Digitais. XVIII Conferência Internacional sobre Informática na Educação, Dezembro. Porto Alegre, p. 9 – 11.
- Veras, W. T. A. e D’Emery, R. A. (2021). “Avaliação de Usabilidade do Jogo Educacional JAvatar”, Relatório Técnico – Universidade Federal Rural de Pernambuco, Serra Talhada, Fevereiro, 20 f.
- Veras, W. T. A., Nogueira, J. E. S. S., Silva, J. E. N e D’Emery, R. A. (2020). “Desenvolvimento e Avaliação da Usabilidade em Aplicativo Desktop para o Ensino da Programação Orientada a Objetos”. In: V Congresso sobre Tecnologias na Educação (Ctrl+E 2020), João Pessoa, Agosto.
- Vieira, S. (2006). “Análise de variância: ANOVA”. São Paulo, Atlas.