Construção e Teste de app gamificado gerador de senhas fortes e memoráveis: Um estudo exploratório em cibersegurança

Hugo Lima Romão[1], Marcelo Henrique Oliveira Henklain[1], Felipe Leite Lobo[1], Eduardo Luzeiro Feitosa[2]

[1]Departamento de Ciência da Computação - Universidade Federal de Roraima – Boa Vista – Roraima – Brasil

[2]Instituto de Ciência da Computação – Universidade Federal do Amazonas – Manaus – Amazonas – Brasil

hugo8romao@gmail.com, {marcelo.henklain, felipe.lobo}@ufrr.br, efeitosa@icomp.ufam.edu.br

Resumo. *Embora o uso de serviços on-line tenha aumentado na última década, a força das senhas criadas pelos usuários não mudou muito. O objetivo deste estudo foi desenvolver e avaliar a eficiência de app gamificado para favorecer o comportamento de "projetar senhas fortes". Participaram, durante nove dias, 10 adultos com média de idade de 22,45 anos. Em comparação com o algoritmo de geração de senhas convencional, verificamos que as senhas geradas por nosso app desempenharam 68,43% melhor em teste de memorização, 4,87% melhor em teste de digitação e 60,38% melhor em teste combinado de memorização e digitação. Nossa abordagem se mostrou promissora na promoção de senhas fortes e memoráveis.*

Abstract. *Although the use of online services has increased in the last decade, the strength of passwords created by users has remained at concerning levels. The aim of this study was to develop and evaluate the efficiency of a gamified app in fostering the behavior of "designing strong passwords". Ten adults with an average age of 22.45 years participated over a nine-day period. Compared to conventional password generation algorithms, passwords generated by our app performed 68.43% better in a memorization test, 4.87% better in a typing test, and 60.38% better in a combined test. Our approach proved to be promising in promoting strong and memorable passwords.*

1. Introdução

O uso da Internet tem se tornado cada vez mais pervasivo [Feldmann, 2021; Wells et al, 2023] e vem acompanhado de desafios para a cibersegurança [Chigada and Madzinga, 2021]. Um deles é o comportamento dos usuários ao projetarem senhas. As senhas são o principal mecanismo de autenticação e proteção de dados, tendendo a continuar assim por tempo indefinido [Bošnjak et al, 2019]. Não obstante, são os componentes da área de cibersegurança mais ignorados, mesmo mediante campanhas de conscientização sobre os riscos de ciberataques [Bošnjak et al., 2018; Carvalho et al., 2017; Ji et al., 2017].

Para contribuir com a atenuação desse problema, definimos como objetivo deste estudo, derivado de TCC, desenvolver e avaliar a eficiência de *app* gamificado para favorecer o comportamento de "projetar senhas fortes". Para tanto, aperfeiçoamos o

algoritmo de geração de senhas de Glory et al. (2019), baseado nas entradas do usuário, a fim de criar senhas fortes e memoráveis. O teste deste *app* envolveu o exame das senhas criadas em termos de facilidade de digitação e memorização em relação a senhas totalmente aleatórias, bem como investigação sobre se o comportamento de projetar senhas fortes foi adquirido ou aperfeiçoado após a utilização do *app*. Avaliamos, ainda, a usabilidade do *app*. Além da introdução, este trabalho possui seções de fundamentação teórica, trabalhos relacionados, método, resultados e discussão e conclusão.

2. Fundamentação Teórica

2.1. Promoção de comportamentos seguros segundo a ótica educacional

O comportamento, termo que abarca fenômenos psicológicos como cognição, sentimentos e atitudes, é definido na Análise do Comportamento como sistema de interações entre ambiente antecedente, ações de um organismo e ambiente subsequente [Skinner, 1981; Kienen et al., 2021]. Projetar senhas fortes é um comportamento cujo ambiente antecedente tem estímulos do tipo “Conhecimento sobre o que caracteriza uma senha como forte”, ações como “Identificar características de uma senha forte" e “Escrever senha” e ambiente subsequente envolvendo “Redução das chances de perda de dados”. Analisar comportamentos dessa forma é útil para o desenvolvimento de recursos pedagógicos, pois fica explícito quais ações devem ser favorecidas, em que condições devem ocorrer e que resultados precisam produzir, para que o comportamento apresentado seja considerado correto e completo [Kienen et al., 2021].

O *app* que construímos é um recurso pedagógico porque visa ensinar pessoas [Cianca et al, 2020], tendo o comportamento de projetar senhas fortes como objetivo de aprendizagem. Esperamos que quem não sabe esse comportamento, após a exposição ao nosso *app*, torne-se apto a emiti-lo, sendo esta mudança o indicador de que o aprendizado ocorreu [Kienen et al., 2021]. Lembramos que o aprendizado decorre da seleção de ações de uma pessoa, em determinado contexto, pelas consequências presentes no ambiente subsequente [Moreira and Medeiros, 2018]. Tais consequências são chamadas de reforçadoras e se caracterizam por efeitos como tornar uma ação mais forte e provável de recorrer [Skinner, 1981; Moreira and Medeiros, 2018]. A proximidade temporal entre ação e consequência é crítica para a ocorrência de aprendizado, assim como a repetição da interação entre ação e consequência reforçadora [Skinner, 1981]. Na prática, pode ser difícil administrar de modo reiterado consequências imediatas para várias pessoas. Por isso, recursos pedagógicos digitais e gamificados podem ajudar os educadores.

2.2. Gamificação

Gamificar significa inserir elementos de jogos em contextos de não-jogo [Azoubel and Pergher, 2017]. A definição clara de objetivos, *feedbacks* imediatos, aumento progressivo de dificuldade e atividades desafiadoras são técnicas de gamificação que mostraram ser úteis para o ensino [Bai et al., 2020]. Um dos motivos dessa efetividade é porque a gamificação usa múltiplas estratégias de reforçamento [Groening and Binnewies, 2019]. Com uma interface agradável e amigável, o comportamento de interagir com o sistema gamificado é favorecido. A indicação de pontos obtidos e desafios superados, por sua vez, aumenta as chances de aprendizado das ações que produziram essas consequências.

Dessa forma, mais do que conscientizar, acreditamos no arranjo de condições que tornem mais provável o comportamento de projetar senhas fortes. Nessa perspectiva, além da gamificação, é útil que a senha gerada seja de fácil memorização.

2.3. Memória interpretada como comportamentos de lembrar e esquecer

A memória pode ser interpretada como um fenômeno comportamental, que envolve os comportamentos de lembrar e esquecer [Skinner, 1981]. Ela consiste na capacidade de uma pessoa ser modificada de modo duradouro (isto é, aprender) e, a partir disso, interagir de um modo novo com o ambiente [Skinner, 1981; Arantes et al., 2012].

O nosso *app* facilita o comportamento de lembrar porque pede a inserção de palavras e números com significado para o usuário. Esses *inputs* têm significado porque podem ser membros de uma classe de estímulos equivalentes (quando, embora distintos, têm o mesmo significado), sendo mais memoráveis quanto maior for o tamanho da classe a que pertencem [Haydu et al., 2009]. Podem também ser estímulos correlacionados por uma cadeia comportamental muito treinada (como uma música, em que a emissão de um verso, torna mais provável lembrar do próximo), o que também os faz mais memoráveis [Arantes et al., 2012]. Por fim, resta, então, esclarecer o que é uma senha forte.

2.4. Criação de senhas fortes

Muitos estudos avaliam o que seria uma senha forte. Ji et al. (2017) examinaram diversos algoritmos de *cracking* de senhas em bases de dados reais e verificaram que senhas com estruturas mais complexas, grande variedade de símbolos e sem dados facilmente identificáveis sobre o usuário são mais difíceis de serem quebradas. Bošnjak et al. (2019) observou, ainda, que o simples fato de uma senha ser muito longa já a torna mais forte. Segundo Glory et al. (2019), o uso de senhas longas é mais fácil para os usuários.

A técnica consagrada na literatura para avaliar essas diferentes políticas, mensurando a força de uma senha, é o cálculo de sua entropia, dado pela relação: *Comprimento_da_Senha x log2(Tamanho_Conjunto_Caracteres)*. Senhas fortes são aquelas cuja entropia é superior a 60 bits [Glory et al., 2019].

3. Trabalhos Relacionados

Conduzimos uma revisão da literatura para demonstrar a novidade deste estudo. Švábenksý et al. (2020) verificaram, em revisão sobre educação em cibersegurança, que o ensino sobre criação de senhas fortes foi pouco abordado. Carvalho et al. (2017), por sua vez, observaram que 40% dos entrevistados em pesquisas sobre cibersegurança não dominavam estratégias de criação de senhas seguras.

Concomitantemente, estudos como o de Abdrabou et al. (2021) mostraram que a criação de senhas fortes, comparadas às fracas, requer alto custo cognitivo do usuário, desestimulando a adoção de senhas seguras. Nesse contexto, Bonk et al. (2021) identificaram que políticas que enfatizam o uso de senhas longas, desde que sejam fáceis de digitar, favorecem a criação de senhas fortes, estratégia recomendada por Han et al (2021) e Mukherjee et al. (2023) como segura e de menor custo cognitivo. Wu et al. (2022) sugerem que senhas longas, formadas por cinco palavras, com três a cinco caracteres cada, são seguras, mas não se mostraram memoráveis. Em conjunto, essas pesquisas sugerem que é preciso propor políticas de segurança com elevada usabilidade e que gerem senhas memoráveis e fáceis de digitar.

Para lidar com esse problema, Glory et al. (2019) criaram algoritmo de geração de senhas que utiliza palavras-chave fornecidas pelo usuário e produz senhas seguras e memoráveis, superando entropia e usabilidade de senhas criadas por geradores de senhas convencionais. Esse algoritmo defende o usuário contra ataques de força bruta porque gera senhas longas e com caracteres variados, e protege contra ataque de dicionário porque usa mais de uma palavra e, ainda, com transformações de caracteres de texto em especiais e inserções de números. Embora promissor, nesse trabalho não foi construída aplicação para usuários reais, tampouco foi avaliado se as senhas geradas de fato são memoráveis e, por fim, não foi pensada uma estratégia para incentivar o uso desse algoritmo pelas pessoas. Decidimos, então, aperfeiçoar o algoritmo de Glory et al. (2019) para desenvolver e testar um *app* gerador de senhas fortes e memoráveis, baseadas em *inputs* do usuário, cujo uso e criação de senhas fortes são favorecidos pela gamificação, contexto que pode ser, especialmente, útil com jovens [Farias et al., 2019]. Esse *app* reduz o custo cognitivo de criação de senhas fortes porque gera a senha, realiza validações de que o usuário precisa para essa tarefa e, ainda, fornece dicas de cibersegurança.

4. Método

4.1. Solução computacional

Desenvolvemos um gerador de senhas gamificado, denominado EasyGuard, que cria senhas fortes, seguindo a política de criação de senhas longas e com caracteres variados. Este *app* pede que o usuário utilize palavras ou frases e dígitos significativos, para que a senha seja memorável, destacando que devem ser evitadas informações óbvias sobre o usuário. Para atenuar esse tipo de problema, verificamos o uso de palavras comuns nas senhas, para evitá-las. Finalmente, criamos um ambiente gamificado para incentivar o uso do *app* e a criação de senhas fortes.

4.1.1. Algoritmo de geração de senhas

O nosso algoritmo (ver Anexo 1), primeiramente, pede para que o usuário insira, no mínimo, quatro palavras e dois números, embora possa inserir mais que isso. As palavras devem ser significativas e de contextos variados. A interface do *app* sugere que as palavras formem uma história (e.g., "buriti, carmesim, corajoso, fez vestibular"). Dessa forma, a senha pode ser memorizada, sem que sejam utilizados termos óbvios relacionados ao usuário. Em seguida, avançando em relação a Glory et al. (2019), o algoritmo verifica se os *inputs* atendem aos seguintes critérios: (1) não podem ser menores que três caracteres ou se repetir; (2) não devem formar padrões de *layout* do teclado (ex.: QWERTY); (3) não devem estar na nossa lista de palavras comuns do Português.

O algoritmo, então, concatena as palavras fornecidas, formando uma frase. Em decisão aleatória, as palavras podem ser separadas por símbolos especiais aleatórios ou um dos números fornecidos pelo usuário. Logo, uma mesma sequência de palavras pode gerar senhas distintas, aumentando a sua resistência, conforme sugerem Han et al. (2021) e Wu et al. (2022). Na última etapa, aplicamos a técnica *LeetSpeak* a um tipo de caractere, escolhido aleatoriamente (e.g., exemplo com a letra i: a senha *buriti\#carmesim.fez-vestibular* se torna *bur!t!\#carmes!m.fez-vest!bular*). Com isso, mesmo que parte das palavras sejam comprometidas, o espaço de busca da senha correta ainda é amplo.

O algoritmo realiza, então, validações adicionais, inovando novamente em relação a Glory et al. (2019): (1) Calcula a entropia da senha gerada, cujo valor mínimo aceitável

é de 60 bits; (2) Verifica se a senha possui, no mínimo, 16 caracteres, requisito que aumenta o custo de adivinhação [Wu et al., 2022; Mukherjee et al., 2023]; (3) Utiliza duas ferramentas *online* para avaliação de segurança. No site *Have I Been Pwned*, que possui vazamentos do Brasil, verificamos se a senha gerada já não foi comprometida. Depois, avaliamos a força da senha no site *The Password Meter*, comumente empregado na literatura [Yildirim and Mackie, 2019; Glory et al., 2019]. Essas ferramentas são projetos de código aberto e a comunicação com elas foi feita a partir de *web scraping*.

Caso a senha gerada não satisfaça qualquer um desses critérios, o *app* pede que sejam fornecidas mais palavras como entrada ou que as informações atuais sejam alteradas por palavras ou dígitos maiores e mais complexos. É importante ressaltar que todo o processo de geração de senha ocorre no dispositivo do usuário, e nada é armazenado de forma permanente, seja pelo *app* ou pelas ferramentas de terceiros que utilizamos. Ademais, a conexão entre o gerador de senhas e as ferramentas online é criptografada por meio do protocolo HTTPS.

4.1.2. Ambiente gamificado

A gamificação empregada no aplicativo foi focada na criação de: (1) tutorial interativo, para reforçar os comportamentos de interagir com o aplicativo; e (2) conquistas, *badges* e *feedback* sobre a força da senha, para reforçar o comportamento de projetar senhas fortes. A seguir, descrevemos o funcionamento desses elementos.

(1) Tutorial interativo. Apresenta o *app*, suas funcionalidades e como utilizá-lo no primeiro acesso, mas pode ser visto novamente, caso o usuário precise. Avaliamos que o ambiente de um *app* recém-instalado pode assumir valor aversivo para o usuário, uma vez que ele ainda precisará aprender a utilizá-lo [Sidman, 1995]. Por isso, este tutorial indica os comportamentos esperados e os reforça quando emitidos. **(2) Conquistas**. Adotamos o design proposto por Groening et al. (2019), para os quais a conquista é consequência informativa sobre o sucesso do usuário em uma tarefa, reforçando os comportamentos que precisam ser aprendidos. Utilizamos dois tipos de conquistas: aquelas relacionadas (i) à utilização das senhas, e (ii) ao uso diário do *app*, totalizando 10 conquistas. Ao completá-las, o usuário recebe como recompensa uma *badge*. Para atingir as conquistas, o usuário deve, na página de conquistas, marcá-la como completa ou, no caso das conquistas que envolvem utilização diária, criar senhas com o *app* pelo máximo de dias consecutivos possível. Importa incentivar a criação de senhas para o aprendizado desse comportamento e para que senhas fracas que o usuário possua sejam substituídas. **(3) *Feedback* em tempo real**. Após informar os *inputs*, é possível copiar a senha gerada ou editá-la. Para evitar que o usuário edite a senha, deixando-a menos segura, é apresentada uma mensagem de *feedback* que se altera conforme o valor de entropia da senha. Caso a entropia fique menor que 60 bits, a função de copiar senha é desabilitada e a mensagem de *feedback* alerta que a senha é fraca.

4.1.3. Tecnologias para o desenvolvimento do programa

Neste estudo, algoritmo de geração de senhas, testes automatizados e rotinas de comunicação com os serviços de avaliação de senha foram feitos com TypeScript, uma extensão de JavaScript, com suporte a tipagem estática [Hejlsberg and Microsoft, 2012]. Foi utilizada também a biblioteca ReactJS, para o desenvolvimento de interfaces [Walke and Facebook, 2013]. Exemplos de telas podem ser vistas no Anexo 2 e o código da

aplicação está disponível no GitHub < https://github.com/hugoromao/easyguard >. O nosso *app* para Android está disponível no endereço < https://bit.ly/45BVQER >.

4.2. Participantes

Participaram 10 adultos com média de idade de 22,45 anos (*DP* = 1,72), variando de 21 a 26. P1, P6, P7, P8, P9, P10 são estudantes de Computação, P2 de Artes Visuais e P5 de Arquitetura. P3 é profissional de computação, na área de interfaces, e P4 é administrador de empresa. Notamos que 30% dessa amostra nunca atualiza as senhas, 30% compartilham-nas às vezes, 70% nunca verificam se suas senhas foram encontradas em vazamentos, 30% sempre guardam suas senhas de forma insegura (e.g., bloco de notas) e 40% fazem isso às vezes; 70% usam a mesma senha regularmente ou sempre e apenas 20% consideram que sempre usam senhas fortes. Além desses participantes, contamos com o auxílio de um estudante de Psicologia no estudo piloto, nos ajudando a avaliar a aperfeiçoar o procedimento de coleta de dados. Todos os participantes assinaram Termo de Consentimento Livre e Esclarecido (TCLE). Esta pesquisa foi aprovada por Comitê de Ética em Pesquisa, Parecer n. 6.668.056, CAAE n. 77411823.6.0000.5302.

4.3. Instrumentos

A exceção do Teste de Usabilidade, os instrumentos descritos a seguir foram desenvolvidos para esta pesquisa. Eles ainda não contam com estudos psicométricos.

Teste de Aprendizado (Pré e Pós-teste). Solicitamos no início da pesquisa e ao final que cada participante realizasse a seguinte tarefa "Crie uma senha que você considere forte e adequada para utilizar no seu dia a dia", para examinarmos eventuais mudanças de extensão, variabilidade de caracteres usados e entropia.

Testes de Custo Cognitivo. (1) Teste de memória: Foram geradas duas senhas aleatórias de 16 caracteres por meio do Bitwarden, sendo fornecido três minutos para que fossem estudadas. Ao final desse tempo, para distração, o participante foi exposto a dois *trailers* de filmes com cinco minutos no total. Na sequência, foi requisitado a digitar as senhas que havia estudado. Ao final dessa atividade, o participante gerou duas senhas com o EasyGuard e o mesmo procedimento foi repetido, mas com novos *trailers*. Analisamos, então, diante de qual tipo de senha o percentual de acertos foi maior. (2) Teste de digitação: Geramos uma nova senha com cada ferramenta e solicitamos que fossem digitadas, uma por vez, o mais rápido possível, repetindo o processo cinco vezes. Esse teste avaliou a frequência de erros de digitação entre as senhas significativas e aleatórias. (3) Teste combinado de memória e digitação: Geramos uma nova senha com cada ferramenta e solicitamos que fosse estudada por três minutos e, de memória, digitada, uma por vez, o mais rápido possível, repetindo o processo cinco vezes. Esse teste avaliou a frequência de erros de digitação, tendo apenas a memória como referência.

Testes de Engajamento. (1) Teste de Usabilidade (ver Anexo 3): Traduzimos o *Post-Study System Usability Questionnaire* (PSSUQ), adotado na literatura [Vlachogianni and Tselios, 2023; Azlan and Junaini, 2023], para a língua portuguesa, ainda sem realizar a adaptação transcultural. Esse instrumento possui 15 itens, emprega uma escala tipo Likert de sete pontos, "1 = Discordo fortemente" a "7 = Concordo fortemente", e seu escore é obtido pela média das pontuações dos seus itens. Uma pontuação mais alta reflete maior concordância em relação à usabilidade. O PSSUQ engloba três subescalas: utilidade do sistema (Itens 1 a 6), qualidade da informação (Itens

7 a 12) e qualidade da interface (Itens 13 a 15). O PSSUQ foi preenchido ao final das tarefas realizadas no laboratório e, opcionalmente, após o nono dia de teste do *app*. (2) Protocolo de registro de uso do gerador de senhas: Permitia registrar o total de senhas criadas e as datas de acesso ao *app* durante os nove dias de uso fora do laboratório.

4.4. Procedimento de coleta e análise de dados

Para aperfeiçoar o método, conduzimos o estudo piloto, por meio do qual decidimos criar o teste combinado e reduzir o tempo de memorização de cinco para três minutos. Dividimos o experimento em duas etapas. Na primeira, em laboratório, os participantes assinaram o TCLE, responderam ao pré-teste, receberam o *link* de acesso ao EasyGuard e, após o tutorial interativo, iniciamos o teste de memória, digitação e combinado. No início de cada teste, foi pedido que os participantes criassem as senhas com o EasyGuard e Bitwarden, que foi escolhido por se tratar de uma solução robusta e de código aberto para gerenciamento de senhas [8Bit Solutions, 2016], cujo algoritmo de geração de senhas reflete a estratégia convencional de senhas totalmente aleatórias. Ao final dos testes, foi pedido aos participantes que respondessem o PSSUQ. Por fim, pedimos que, em casa, tentassem atingir as conquistas disponíveis no EasyGuard ao longo de nove dias, registrando o uso da ferramenta.

Na segunda etapa, realizamos a coleta dos registros de uso do *app* pelos participantes, aplicamos o pós-teste e perguntamos se os participantes tinham interesse em alterar alguma resposta em relação ao PSSUQ. A análise dos dados foi quali-quanti, predominando o uso de estatísticas descritivas. Calculamos o teste de postos sinalizados de Wilcoxon, que é não paramétrico, para comparar resultados de pré e pós-teste e diferenças de desempenho em função do uso do EasyGuard ou Bitwarden. Contudo, os achados foram os mesmos quando utilizado o teste T para amostras pareadas e, por este ser mais robusto, reportamos os seus resultados. Os dados coletados estão disponíveis neste *link* http://dx.doi.org/10.13140/RG.2.2.33371.99362.

5. Resultados e Discussão

5.1. Resultados dos testes de memória, digitação e combinado

As Figuras 1A, 1B e 1C mostram o percentual de acertos nos testes de memória, digitação e combinado para os 10 participantes. A Figura 1D mostra a média de acertos em cada teste em função da aplicação utilizada. Os resultados no teste de memória indicaram maior percentual de acertos com o EasyGuard (*Média* = 92,81; *DP* = 9,79) em comparação ao Bitwarden (*Média* = 24,38; *DP* = 30,61), com tamanho do efeito grande (t = 7,407; gl = 9; p < 0,001; d = 2,342). Tal resultado corrobora a hipótese de Glory et al. (2019) de que o uso de senhas geradas a partir de *inputs* do usuário são mais memoráveis. A única exceção que encontramos foi P2, cujo desempenho foi similar para os dois tipos de senhas. É possível que ele tenha utilizado alguma técnica de memorização. Destacamos que P5 a P10 reportaram ter esquecido as senhas aleatórias após o período de distração, sendo que esse baixo percentual de acertos exibido sugere desatenção. Esse dado ilustra que, em estudos futuros, seria útil perguntar como o participante se organizou para memorizar as senhas, bem como seria importante realizar o teste após um dia ou mais de criação da senha, o que representaria melhor a realidade.

No teste de digitação, avaliamos o custo associado ao uso das senhas geradas pelas duas ferramentas, uma vez que senhas mais difíceis de digitar são mais custosas [cf. Bonk

et al., 2021]. Encontramos uma diferença menor e não significativa (t = 1,410; gl = 9; p = ,0192) entre as duas abordagens, com 92% (DP = 4,44) de acertos usando o EasyGuard e 87,13% (DP = 9,88) com o Bitwarden. Embora as senhas geradas pelo nosso *app* tendam a ter menor custo de digitação, pois são compostas por termos conhecidos do usuário [Shay et al., 2014; Bonk et al., 2021], o fato de que não controlamos o tempo para digitar, total de correções ao digitar e permitimos consulta às senhas geradas, dificultou a identificação de uma diferença. Esses são aspectos a melhorar e que tentamos aperfeiçoar com a inclusão do teste combinado. Com ele, encontramos dados favoráveis ao EasyGuard. A média de acertos foi de 92,38% (DP = 4,05) em nosso *app* e de 32% (DP = 38,28) no Bitwarden, sendo essa diferença significativa e com tamanho do efeito grande (t = 4,599; gl = 9; p < 0,001; d = 1,454). Novamente, P5 a P10 relataram maior dificuldade durante a recuperação das senhas geradas pelo Bitwarden, alegando que ora se esqueciam da senha após os *trailers*, ora a confundiam com outras criadas anteriormente. Ao mesmo tempo, os participantes P1 a P4 apresentaram pouca diferença entre as abordagens. Logo, a ampliação da amostra e o aperfeiçoamento dos testes são necessários em estudos futuros.

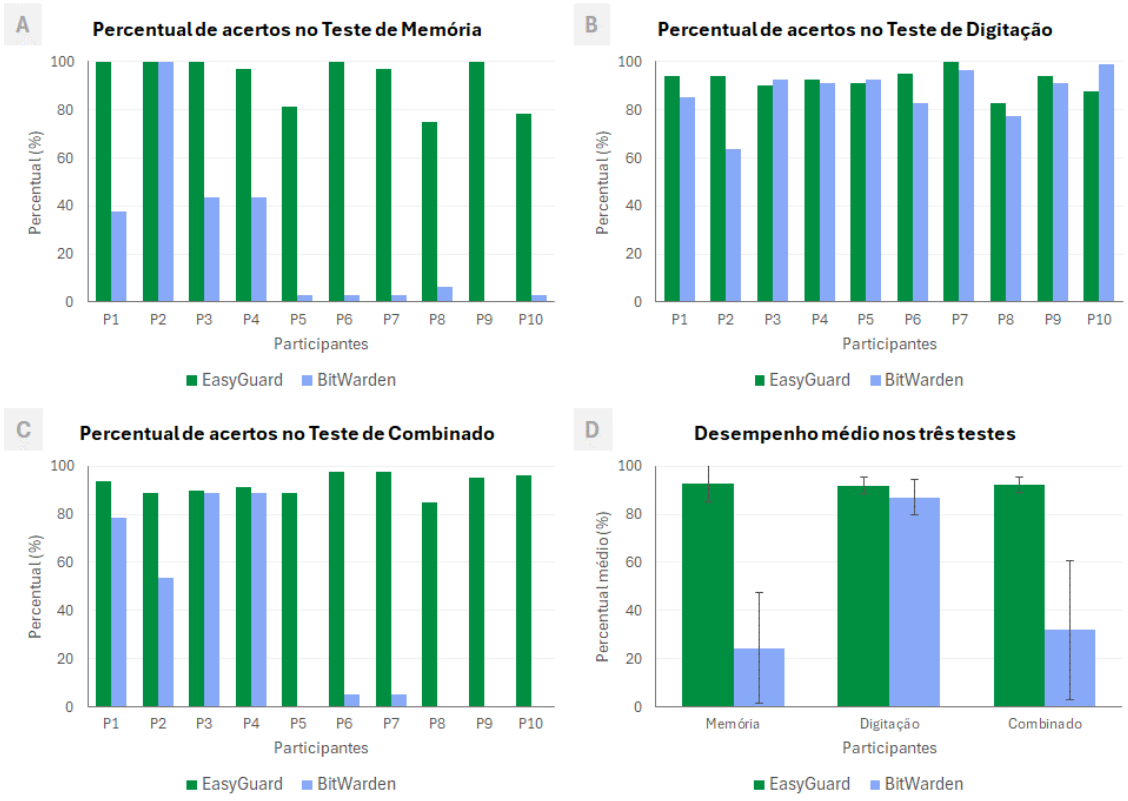


Figura 1. Percentuais de acertos nos testes de memória (Figura 1A), digitação (Figura 1B) e combinado (Figura 1C), e percentual de acertos médio em todos os testes (Figura 1D).

Em síntese, os resultados dos testes sugerem que senhas geradas com o EasyGuard são mais memoráveis que as totalmente aleatórias, além de terem maior usabilidade, no sentido de maior facilidade para digitar. Esse é um achado promissor, pois Wu et al. (2022) verificaram que, no seu método de senhas longas baseadas em cinco palavras, seus participantes não conseguiram lembrar as senhas que criaram, em teste realizado após duas semanas. Examinamos a seguir como os participantes avaliaram a usabilidade do *app*, indicador crucial sobre se ele tende a ser utilizado em um cenário real pelos usuários.

5.2. Resultado do teste de usabilidade

A Figura 2A exibe os escores médios de usabilidade por participante e em função das subescalas de utilidade, informação e interface. A Figura 2B mostra o escore médio nessas três subescalas. Podemos notar altos escores nas três subescalas. No geral, a média foi de 6,77 em utilidade (*DP* = 0,25), 6,52 em informação (*DP* = 0,40) e 6,97 para interface (*DP* = 0,10). A menor média de utilidade foi 6,17, de informação foi 5,83 e de interface foi 6,67, ou seja, todas altas, sugerindo que o EasyGuard foi percebido como fácil de ser utilizado, principalmente, no que tange à interface. Não obstante, acreditamos que ainda devemos aperfeiçoar a dimensão de informação porque, durante a aplicação dos testes, surgiram dúvidas sobre o funcionamento da ferramenta (e.g., como atingir as conquistas, quantidade de palavras e números que poderiam ser inseridos). Segundo Sauro e Lewis (2012), isso é um indicativo de que os recursos da aplicação precisam ser melhorados. Em seguida, examinaremos o uso do *app* fora do laboratório.

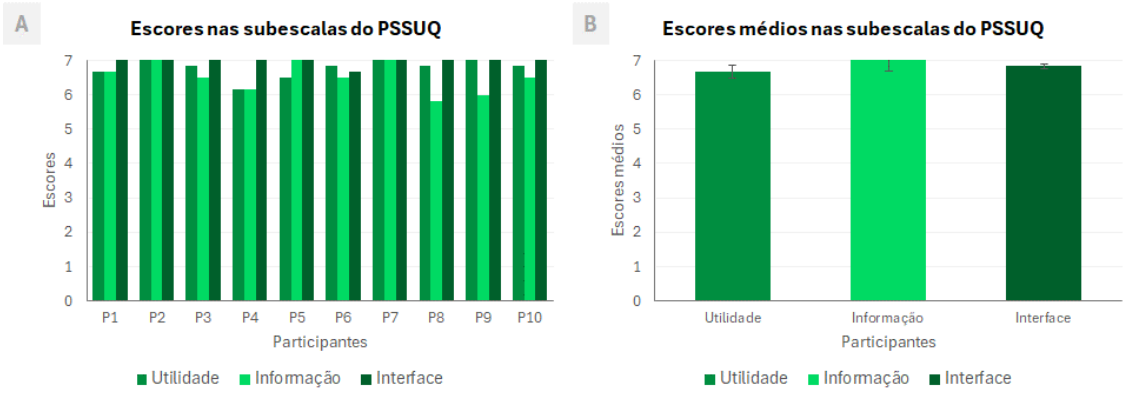


Figura 2. Escore médio das subescalas do PSSUQ para cada participante (Figura 2A) e escore médio de todos os participantes (Figura 2B).

5.3. Resultado do Protocolo de Registro de Uso

Durante o período de uso do *app*, os participantes foram instruídos a registrar a quantidade de senhas criadas e conquistas atingidas a cada dia. A quantidade de uso do *app* é uma medida direta do engajamento ao comportamento de "projetar senhas fortes".

Em média, os participantes utilizaram o *app* por 1,60 dias (*DP* = 1,28). P6 foi o que mais utilizou o *app* e P5 o que mais criou senhas (9) e alcançou conquistas (9). Em contrapartida, P7 e P9 não utilizaram o *app* fora do laboratório. As conquistas parecem ter desempenhado função reforçadora apenas para P2, P3, P5 e P6, pois foram os participantes que mais criaram senhas. Resultados semelhantes foram encontrados por Groening et al. (2019), pois um dos efeitos do uso de conquistas foi o aumento do nível de persistência. No entanto, analisando o uso de P1 e P10, percebemos uma desistência logo após o primeiro dia. Conforme esperado pela teoria, um mesmo estímulo não necessariamente terá o mesmo valor para diferentes pessoas, podendo ser reforçador para uma e não para outra [Moreira and Medeiros, 2018]. Isso nos conduz à necessidade de mais estudos, afinal, em face de um comportamento de alto custo, como o de criação de senhas fortes, e cujas consequências reforçadoras ou punitivas são atrasadas temporalmente, precisamos continuar investigando sobre como reforçar a criação de senhas fortes, aperfeiçoando a gamificação presente em nossa ferramenta. Por fim, cumpre verificar se houve algum grau de aprendizado decorrente do uso do EasyGuard.

5.4. Resultados do teste de aprendizado (Pré e Pós-teste)

Verificamos que, antes da intervenção, as senhas criadas tinham em média 10,90 caracteres (*DP* = 2,21), 3,70 grupos de caracteres diferentes (*DP* = 0,64) e 68,06 bits de entropia (*DP* = 16,02). Após o uso do *app*, encontramos a média de 15,40 caracteres (*DP* = 4,84) por senha, com diferença estatisticamente significativa e tamanho do efeito elevado (t = 2,645; gl = 9; p = 0,027; d = 0,836). Notamos a manutenção da variabilidade de caracteres, mas o crescimento da entropia para 96,82 bits (*DP* = 32,21), sendo também uma diferença estatisticamente significativa com elevado tamanho do efeito (t = 2,668; gl = 9; p = 0,026; d = 0,844). Esses resultados mostram uma mudança comportamental na forma como os usuários criam senhas, ocorrida após a exposição ao *app*, sugerindo que o comportamento de projetar senhas fortes foi aperfeiçoado e, portanto, o *app* EasyGuard foi eficiente enquanto recurso pedagógico [Kienen et al., 2021].

5.5 Limitações do estudo

Este estudo possui limitações que devem ser consideradas em estudos futuros. Sobre o algoritmo, nossa abordagem favorece a proteção contra ataques de força bruta e dicionário, mas ainda não está imune a ataques, por exemplo, de *shoulder surfing* ou direcionados a um usuário. Além disso, precisamos avaliar o EasyGuard a partir de instrumentos com evidências psicométricas e coletar dados com mais participantes, cujas idades sejam mais diversas, e com maior tempo de uso do *app*, de modo a dispor de dados mais robustos sobre a sua efetividade. Para isso, automatizar e refinar o processo de coleta é essencial para a aplicação dos instrumentos em larga escala com usuários reais, aspecto que começamos a desenvolver para um novo estudo. Adicionalmente, é preciso testar esse *app* em relação a outras ferramentas, além do Bitwarden, contemplando, pelo menos, uma que se proponha a gerar senhas memoráveis, como o algoritmo de Glory et al. (2019).

6. Conclusão

O objetivo deste estudo foi desenvolver e avaliar a eficiência de *app* gamificado para a promoção do comportamento de "projetar senhas fortes". Criamos o *app* para que funcionasse como ambiente reforçador e que atenuasse o esforço de criação de senhas fortes. Verificamos que o uso de entradas significativas em combinação com elementos gamificados se mostraram promissores no aperfeiçoamento desse comportamento. Os resultados sugerem que senhas geradas pelo EasyGuard são mais memoráveis e possuem menor custo cognitivo em comparação com a abordagem usual de gerar senhas totalmente aleatórias. Sugerimos que estudos futuros avancem em relação a esta pesquisa, buscando lidar com as limitações que apresentamos. É essencial aperfeiçoar os elementos gamificados desenvolvidos para aumentar a adesão ao *app* e encontrar mais reforçadores para o comportamento de projetar senhas fortes. Estudos futuros podem avaliar métodos para fortalecer as senhas contra outros tipos de ataques.

Este estudo, embora exploratório, produziu resultados promissores, favorecendo o uso do EasyGuard. Inovamos neste trabalho, principalmente, porque avançamos em relação ao algoritmo de criação de senhas baseado em *inputs* do usuário, desenvolvemos um artefato tecnológico a partir desse algoritmo e o testamos em um cenário realista, tendo orientado a nossa pesquisa por teoria psicológica sólida. Finalmente, disponibilizamos o nosso *app* e dados coletados para uso e escrutínio público.

7. Referências

8bit Solutions. (2016). Bitwarden Open-Source Password Manager. Disponível em: < https://bitwarden.com >. Acesso em: 23 de novembro de 2023.

Abdrabou, Y., Abdelrahman, Y., Khamis, M., and Alt, F. (2021). Think harder! Investigating the effect of password strength on cognitive load during password creation. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems (CHI EA '21) (pp. 1-7)*. New York: ACM. https://doi.org/10.1145/3411763.3451636

Arantes, A. K. L., Mello, E. L., and Domeniconi, C. (2012). Memória. In M. M. C. Hübner and M. B. Moreira (Orgs.), *Temas clássicos da psicologia sob a ótica da análise do comportamento (pp. 56-73)*. Rio de Janeiro: Guanabara Koogan.

Azlan, Z. H. Z., and Junaini, N. S. (2023). Erudite Survivor: Usability testing of a gamification-based mobile app for disaster awareness among children. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *31*(3), 290-298. https://doi.org/10.37934/araset.31.3.290298

Azoubel, M. S., and Pergher, N. K. (2017). Levantamento sobre a utilização de jogos na Análise do Comportamento Aplicada. *Perspectivas em Análise do Comportamento*, *8*(2), 215-225. https://doi.org/10.18761/PAC.2016.014

Bai, S., Hew, F. K., and Huang, B. (2020). Does gamification improve student learning outcome? Evidence from a meta-analysis and synthesis of qualitative data in educational contexts. *Educational Research Review*, *30*, 100322. https://doi.org/10.1016/j.edurev.2020.100322

Bonk, C., Parish, Z., Thorpe, J., and Salehi-Abari, A. (2021). Long passphrases: Potentials and limits. In *18th International Conference on Privacy, Security and Trust (PST) (pp. 1-7)*. New York: IEEE. https://doi.org/10.1109/PST52912.2021.9647800

Bošnjak, L., and Brumen, B. (2019). Rejecting the death of passwords: Advice for the future. *Computer Science and Information Systems*, *16*(1), 313-332. https://doi.org/10.2298/CSIS180328016B

Bošnjak, L., Sreš, J., and Brumen, B. (haha). Brute-force and dictionary attack on hashed real-world passwords. In *41st International Convention on Information and Communication Technology (pp. 1161-1166)*. New York: IEEE. https://doi.org/10.23919/MIPRO.2018.8400211

Carvalho, E. A., Reis, T., and Alves, F. J. (2017). Ensino de noções básicas de segurança da informação nas escolas brasileiras. In *Workshop de Informática na Escola (WIE) (pp. 765-774)*. Porto Alegre: Sociedade Brasileira de Computação. https://doi.org/10.5753/cbie.wie.2017.765

Chigada, J., and Madzinga, R. (2021). Cyberattacks and threats during COVID-19: A systematic literature review. *South African Journal of Information Management*, *23*(1), a1277. https://doi.org/10.4102/sajim.v23i1.1277

Cianca, B. C., Panosso, M. G., and Kienen, N. (2020). Programação de Condições para Desenvolvimento de Comportamentos: Caracterização da produção científica brasileira de 1998-2017. *Perspectivas em Análise do Comportamento*, *11*(2), 114–136. https://doi.org/10.18761/PAC.2020.v11.n2.01

Farias, O. L. F., Medeiros, A. A. N., Rocha, L. S., Medeiros, F. D., Nóbrega, C. E., Burlamaqui, F. M. A., and Madeira, G. A. C. (2019). Self protect: Um jogo para auxílio no ensino de conceitos relacionados a segurança na internet para crianças e adolescentes. In *Workshop de Informática na Escola (pp. 246-255)*. Porto Alegre: SBC. https://doi.org/10.5753/cbie.wie.2019.246

Feldmann, A. (2021). A year in lockdown: How the waves of COVID-19 impact internet traffic. *Communications of the ACM*, *64*(7), 101-108. https://doi.org/10.1145/3465212

Glory, Z. F., Aftab, U. A., Tremblay-Savard, O., and Mohammed, N. (2019). Strong password generation based on user inputs. In *IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 416-423)*. New York: IEEE. https://doi.org/10.1109/IEMCON.2019.8936178

Groening, C., and Binnewies, C. (2019). "Achievement unlocked!": The impact of digital achievements as a gamification element on motivation and performance. *Computers in Human Behavior*, *97*, 151-166. https://doi.org/10.1016/j.chb.2019.02.026

Han, W., Xu, M., Zhang, J., Wang, C., Zhang, K., and Wang, X. S. (2021). TransPCFG: Transferring the grammars from short passwords to guess long passwords effectively. *IEEE Transactions on Information Forensics and Security*, *16*, 451-465. https://10.1109/TIFS.2020.3003696

Haydu, V. B., Omote, L. C. F., Vicente, P., Ággio, N. M., and De Paula, J. B. C. (2009). Efeitos do tamanho da classe na manutenção de relações de equivalência em um delineamento intragrupo. *Interação em Psicologia*, *13*, 179-193.

Hejlsberg, A., and Microsoft. (2012). TypeScript. Disponível em: < https://www.typescriptlang.org >. Acesso em: 23 de novembro de 2023.

Ji, S., Yang, S., Hu, X., Han, W., Li, Z., and Beyah, R. (2017). Zero-sum password cracking game: A large-scale empirical study on the crackability, correlation, and security of passwords. *IEEE Transactions on Dependable and Secure Computing*, *14*(5), 550-564. https://doi.org/10.1109/TDSC.2015.2481884

Kienen, N., Panosso, M. G., Nery, A. G. S., Waku, I., and Carmo, J. S. (2021). Contextualização sobre a programação de condições para desenvolvimento de comportamentos (PCDC): Uma experiência brasileira. *Perspectivas em Análise do Comportamento*, *12*(2), 360-390. https://doi.org/10.18761/PAC.2021.jul110

Moreira, B. M., and Medeiros, A. C. (2018). *Princípios básicos de análise do comportamento*. Porto Alegre: Artmed.

Mukherjee, A., Murali, K., Jha, K. S., Ganguly, N., Chatterjee, R., and Mondal, M. (2023). MASCARA: Systematically generating memorable and secure passphrases. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security (pp. 524-538)*. New York: ACM. https://doi.org/10.1145/3579856.3582839

Sauro, J., and Lewis, R. J. (2012). *Quantifying the user experience:* Practical statistics for user research. Waltham: Elsevier.

Shay, R., Komanduri, S., Durity, L. A., Huh, P., Mazurek, L. M., Segreti, M. S., Ur, B., Bauer, L., Christin, N., and Cranor, F. L. (2014). Can long passwords be secure and usable? In *CHI 14 Conference on Human Factors in Computing Systems (pp. 2927-2936)*. Nova Iorque: ACM. https://doi.org/10.1145/2556288.2557377

Sidman, M. (1995). *Coerção e suas implicações*. Campinas: Editorial Psy.

Skinner, B. F. (1981). *Ciência e comportamento humano*. São Paulo: Martins Fontes.

Švábenksý, V., Vykopal, J., and Čelada, P. (2020). What are cybersecurity education papers about? A systematic literature review of SIGCSE and ITiCSE conferences. In *51st ACM Technical Symposium on Computer Science Education (pp. 2-8)*. New York: ACM. https://doi.org/10.1145/3328778.3366816

Vlachogianni, P., and Tselios, N. (2023). Perceived usability evaluation of educational technology using the post-study system usability questionnaire (PSSUQ): A systematic review. *Sustainability*, *15*(17). https://doi.org/10.3390/su151712954

Walke, J., and Facebook. (2013). *ReactJS*. Disponível em: < https://react.dev >. Acesso em: 23 de novembro de 2023.

Wells, J., Scheibein, F., Pais, L., Santos, R. N., Dalluege, C., Czakert, P. A., and Berger, R. (2023). A systematic review of the impact of remote working referenced to the concept of work-life flow on physical and psychological health. *Workplace Health Saf*, *71*(11), 507-521. https://doi.org/10.1177/21650799231176397

Wu, X., Munyendo, W. C., Cosic, E., Flynn, A. G., Legault, O., and Aviv, J. A. (2022). User perceptions of five-word passwords. In *Annual Computer Security Applications Conference (pp. 605-618)*. New York: ACM. https://doi.org/10.1145/3564625.3567981

Yildirim, M., and Mackie, I. (2019). Encouraging users to improve password security and memorability. *International Journal of Information Security*, *18*(6), 741-759. https://doi.org/10.1007/s10207-019-00429-y

8. Anexos

Anexo 1 – Algoritmo de geração de senhas.

Algorithm 1 Algoritmo de geração de senhas

Require: Quatro palavras e dois números significativos para o usuário.

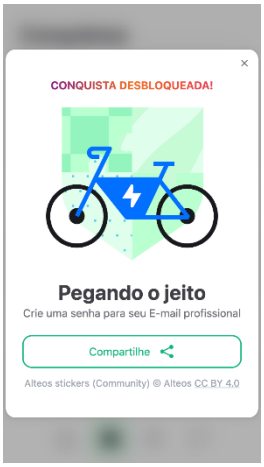
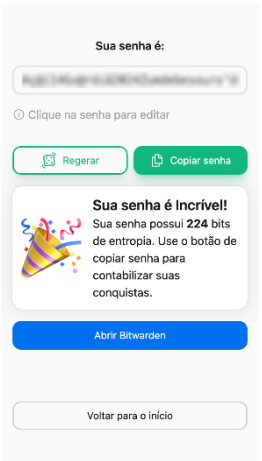
```
palavras ← [buriti, carmesim, corajoso, fez vestibular]
numeros ← [43, 92]
especiais ← [−, @, ∗, =, ., +, ;, /, (, ), !]
wordlist ← lista de palavras comuns
keyboardlist ← lista de padrões de teclado

arr ← []
for i ← 1, palavras.tamanho do
    if arr.inclui(palavras[i]) then                ▷ Requisito: Palavras não devem se repetir
        Falha: "A palavra " + palavras[i] + " está sendo repetida."
    end if
    if palavras[i].tamanho < 3 then                ▷ Requisito: Mínimo 3 caracteres
        Falha: "A palavra " + palavras[i] + " é muito curta."
    end if
    if keyboardlist.include(palavras[i]) then      ▷ Requisito: Sem padrões de teclado
        Falha: "A palavra " + palavras[i] + " inclui um padrão de teclado."
    end if
    if wordlist.include(palavras[i]) then          ▷ Requisito: Não utilizar palavras comuns
        Falha: "A palavra " + palavras[i] + " é muito comum."
    end if
    arr.add(palavras[i])
end for

senha ← palavras.concatenar(pegarItemAleatorio(especiais, numeros))
letras ← senha.separarCaracteres()
letra ← pegarItemAleatorio(letras)
senha ← senha.substituir(letra, pegarAleatorio(especiais, numeros))   ▷ técnica LeetSpeek
```

if $senha.tamanho * log(contarTiposDeCaracteres(senha), 2) < 60$ **then**
    Falha: "A senha gerada não possui o nível mínimo de entropia. Recomendamos adicionar mais palavras ou substituir as atuais por opções mais complexas."
end if
if $senha.tamanho < 16$ **then**
    Falha: "A senha gerada é muito curta. Recomendamos adicionar mais palavras ou substituir as atuais por opções mais complexas."
end if
if $avaliacaoExterna(senha)$ *igual* $Falso$ **then**
    Falha: "A senha gerada não passou na avaliação de segurança de ferramentas online. Utilize palavras menos comuns e que não estejam presentes em qualquer outra senha que você já utilize."
end if
return senha

Anexo 2 – Exemplos de telas do *app* EasyGuard.

A	B	C	D
			

Anexo 3 – Instrumento de mensuração de usabilidade (PSSUQ).

AVALIAÇÃO DE PERCEPÇÃO DE USABILIDADE
Escala: 1 - Discordo Fortemente a 7 - Concordo Fortemente **01)** No geral, estou satisfeito com o quão fácil é usar este sistema. **02)** Foi simples usar este sistema. **03)** Consegui concluir as tarefas rapidamente usando este sistema. **04)** Eu me senti confortável usando este sistema. **05)** Foi fácil aprender a usar este sistema. **06)** Eu acredito que eu poderia me tornar produtivo rapidamente usando este sistema. **07)** O sistema apresentou mensagens de erro que claramente indicaram como corrigir problemas. **08)** Sempre que cometi algum erro, eu pude recuperar de forma fácil e rápida. **09)** As informações (como tutoriais, mensagens na tela e outras documentações) fornecidas com este sistema eram claras. **10)** Foi fácil encontrar as informações que eu precisava. **11)** As informações foram eficazes para me ajudar a concluir as tarefas e cenários. **12)** A organização das informações nas telas do sistema estava clara. **13)** A interface do sistema é agradável. **14)** Eu gostei de usar a interface deste sistema. **15)** No geral, estou satisfeito com este sistema.