

sandbox

CÓDIGO | DESIGN | PROCESSOS | CARREIRA



Revista “**Sandbox**”. Edição 005, janeiro de 2025. “Sandbox” é uma revista de distribuição gratuita. Textos e ilustrações pertencem aos seus autores e não podem ser distribuídas fora da revista, alteradas ou vendidas sem autorização. Para mais informações ou para solicitar licença sobre conteúdo, entrar em contato com academy@lds.ifce.edu.br.

No momento da escrita das matérias desta revista, as versões mais recentes do Xcode e Swift eram 16 e 6.0 respectivamente. Consulte o histórico de atualizações caso esteja trabalhando com uma versão diferente dessas.

Saiba mais sobre o
Apple Developer Academy IFCE em:
www.developeracademy.ifce.edu.br



Equipe de Redação

João Roberto, Felipe Frances, Lucas Baptista, Maria Clara, Maria Eduarda, Melissa Freire, Paulo Henrique Melo

Equipe de Revisão

Kelly Letícia, Felipe Frances, Marcelle Ribeiro, Paulo Henrique Melo

Equipe de Ilustração

Daniel Oppelt, David Augusto, Gilberto Neto, Lucas Baptista, Kamilly Ferreira, Marcelle Ribeiro, Maria Clara, Raissa Parente

Equipe de Redação Colaboradora

Carolina Parente (Former mentor IFCE)
Victor Martins (Junior learner PUC-Rio)

Revisão Técnica de Código

Beatriz Leonel e Mateus Rodrigues

Revisão Técnica de Design

David Augusto, Elis Vieira e Lucas Baptista

Capa

Marcelle Ribeiro

Projeto gráfico

Lucas Baptista

Diagramação

Lucas Baptista

Edição Geral

David Augusto e Lucas Baptista

CONTATO

Equipe de Redação

João Roberto – linkedin.com/in/joão-roberto-72a05b217
Felipe Frances – linkedin.com/in/felipefrancesguimaraes
Maria Clara – linkedin.com/in/maria-clara-de-oliveira-alexandre-173ba8245
Maria Eduarda – linkedin.com/in/maria-eduarda-caetano-721490242
Melissa Freire – linkedin.com/in/melissafreireguedes
Paulo Henrique Melo – linkedin.com/in/paulohenriquemelo
Victor Martins (PUC-RIO) – linkedin.com/in/victor-martins-5187a7232

Equipe de Revisão

Kelly Letícia – linkedin.com/in/kelly-letícia-09b61a263
Felipe Frances – linkedin.com/in/felipefrancesguimaraes
Marcelle Ribeiro – linkedin.com/in/marcellerg
Paulo Henrique Melo – linkedin.com/in/paulohenriquemelo

Equipe de Ilustração

Daniel Oppelt – linkedin.com/in/daniel-oppelt-13995b208
Gilberto Neto – linkedin.com/in/gilberto-neto21
Kamelly Ferreira – kmoraesadm@gmail.com
Marcelle Ribeiro – linkedin.com/in/marcellerg
Maria Clara – linkedin.com/in/maria-clara-de-oliveira-alexandre-173ba8245
Raissa Parente – linkedin.com/in/raissa-parente-70bb25162

Equipe de Mentores

Beatriz Leonel – linkedin.com/in/beatriz-leonel-28b948207
David Augusto – linkedin.com/in/davidaugustoss
Elis Vieira – linkedin.com/in/elisionara
Lucas Baptista – linkedin.com/in/lucsbos
Mateus Rodrigues – linkedin.com/in/mtsrodrigues

Apple Developer Academy

Site – developeracademy.ifce.edu.br
LinkedIn – linkedin.com/company/developeracademy-ifce
Medium – medium.com/developeracademy-ifce
Instagram – instagram.com/developeracademy.ifce

EDITORIAL

TEXTO
MARIA CLARA

"Eu achava que sabia exatamente quem eu precisava ser no Academy. Até que um dia, isso parou de funcionar."

Desde o primeiro dia, eu sabia que precisava aproveitar ao máximo o Academy. Depois de um longo processo seletivo, tudo tinha só começado.

Conhecida por parecer dormir no Academy de tão cedo que chegava, no início tudo ali parecia um sonho, equipamentos e pessoas incríveis para nos apoiar a criar o que quiséssemos. Porém, também foi um período de adaptação: lidar com uma nova rotina, tentar me enturmar com 50 desconhecidos e aprender a trabalhar com uma metodologia ativa.

Para dar conta disso, deixei minha "pessoa profissional" dominar. Ela me ajudou a participar ativamente, levantar a mão no campfire e estar presente em toda atividade extra: Reflection, Proday, Funday, CocoaHeads, SandBox... tudo que é nome em inglês. Nada podia dar errado, né?

Bem, mais ou menos. Embora tenha sido essencial abraçar as oportunidades, percebi que minha motivação consistia em sustentar rigorosamente a pessoa que fui no processo seletivo e me provar merecedora daquele lugar que eu ocupava — para os mentores, meus colegas e, principalmente, para mim mesma. Até que, um dia, isso parou de funcionar.

A energia para manter esse muro era enorme, e quando meu "eu profissional" se esgotou, perdi a vontade de interagir e me dedicar. Foi difícil lidar com essa mudança, mas, ironicamente, quando essa parte de mim enfraqueceu, outros aspectos tiveram espaço para crescer.

A partir desse momento, comecei a enxergar um cantinho para mim ali, mesmo que pequeno. O que me fez entender por que os mentores falavam tanto sobre "lugar seguro", algo que me permitiu, sem muros, me sentir confortável sendo eu mesma.

Assim, o Academy se tornou uma experiência muito mais engrandecedora para mim. Ainda participo de muita coisa (olha eu aqui no Editorial!), mas agora de forma mais leve, permitindo que todas as minhas facetas cresçam juntas.

Agradeço muito a todos que me acolheram como partes essenciais desse recomeço feliz!

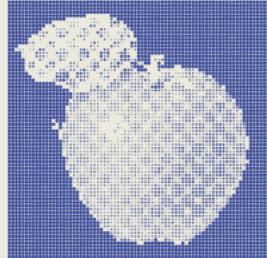
E a quem já se sentiu deslocado, desejo que encontre o seu cantinho, buscando algo sempre de dentro para fora. Nunca será tarde demais para isso.

Confesso que trazer esse relato para cá não foi fácil. Eu já tinha um texto impersonal pronto, mas percebi que essa era uma chance de me conectar com você de uma forma diferente. Afinal, a essência do Academy está em suas histórias, desafios e conquistas, como você vai conferir nesta 5ª edição da SandBox.

Reparou na nossa capa? Ela celebra o primeiro challenge usando um Vision Pro feito por um Academy! Será ele o futuro? Confere isso na nossa matéria dedicada ao tema.

E, claro, não deixe de ler as outras matérias! Elas refletem o espírito multidisciplinar do Apple Academy IFCE e espero que possam enriquecer não apenas sua jornada profissional, mas também sua jornada pessoal.

Boa leitura e até a próxima! :)



JORNAL da Maçã

10
anos

de Apple Developer
Academy IFCE

apresenta...

NOSSO ANO

em
números

154

PROTÓTIPOS PARA iOS

23

PROTÓTIPOS
PARA VISIONOS

15
APPS NA LOJA

#FEITO NO
ACADEMY

90 91 92

LEARNERS

1742
DOWNLOADS

IMPRESSÕES
33.732

36 WORKSHOPS
E TALKS

4 FUN
DAYS

10
CHALLENGES
DO ACADEMY

1/2 do
programa
regular

2 turmas
foundation

80 APREN-
DIZADOS

Obrigado
por participar disso!

CONTEÚDO

07

REALIDADE VIRTUAL É REALMENTE
O FUTURO? UMA REFLEXÃO POR
MEIO DOS TRÊS NÍVEIS EMOCIONAIS



10

FEEDBACK: ALGO
DE ERRADO NÃO
ESTÁ CERTO

14

10 HEURÍSTICAS DE NIELSEN
(TAYLOR'S VERSION)



18

ALÉM DOS ZEROS E UNS: ENTENDENDO
AS MÉTRICAS DO MUNDO DIGITAL



24

COMO OS X-MEN UTILIZAM OS
PRINCÍPIOS DE FORTES EQUIPES DE PRODUTO

18

EXPLORANDO A LINGUÍSTICA
COMPUTACIONAL COM O
FRAMEWORK NATURAL LANGUAGE

34

ACESSIBILIDADE NA TERRA DE OOO:
EXPLORANDO O DESIGN INCLUSIVO
ATRAVÉS DE HORA DE AVENTURA

38

ENTENDENDO SWIFT CONCURRENCY

45

VOCÊ É UMA
PEDRA NO SEU
CAMINHO? UMA
CONVERSA SOBRE
AUTOCRÍTICA



DESIGN

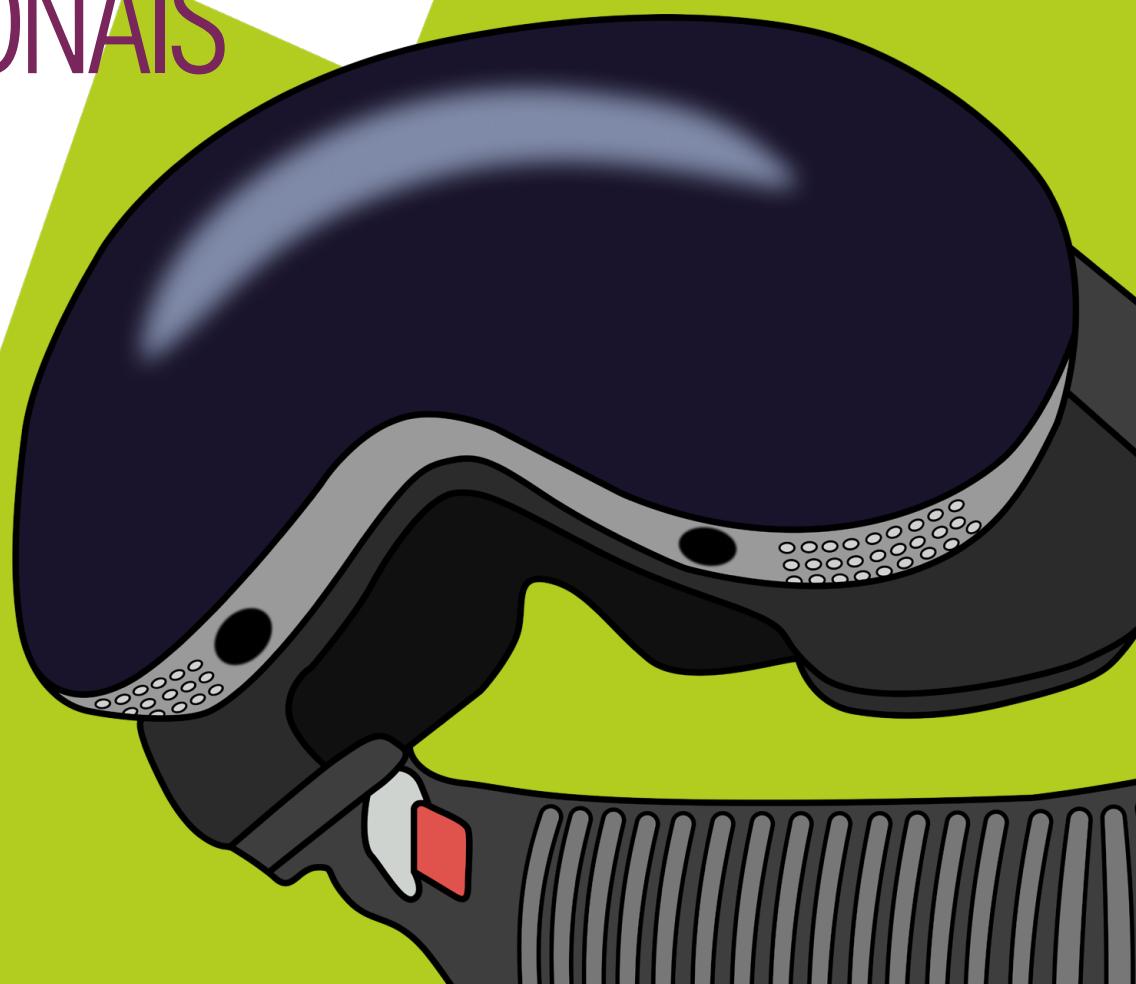
REALIDADE VIRTUAL É REALMENTE O FUTURO?

UMA REFLEXÃO
POR MEIO DOS
TRÊS NÍVEIS
EMOCIONAIS

TEXTO
LUCAS BAPTISTA

ILUSTRAÇÃO
MARCELLE RIBEIRO

REVISÃO
FELIPE FRANCES



Visceral

Londres, 1896. O sol começa a se pôr enquanto o público de um espetáculo teatral encontra seu assento. Todos estão curiosos sobre o roteiro da peça, espectadores que foram em sessões anteriores não conseguiram dormir por noites. O espetáculo começa tímido, com atores no palco e um roteiro de suspense. Em determinado momento da peça, gritos de histeria são ouvidos do interior do teatro. Fantasmas saem do palco e começam a sobrevoar a platéia. Mesmo sendo um jogo de luzes, fumaça e som que simulava o efeito 3D, ainda houve quem saísse dali acreditando no sobrenatural. O efeito "wow" daquela peça criou um fenômeno de bilheteria.

Em 2007, Steve Jobs anuncia um smartphone sem teclado físico, apenas uma tela Multi-Touch e uma câmera traseira – a plateia grita "wow!". Inovações tecnológicas sempre trarão surpresa ao público, é uma das consequências inevitáveis que abrangem o conceito de inovar. Entretanto, o efeito "wow" tem um prazo de validade e ele é muito efêmero. Quando o segredo do truque de mágica é revelado ou você o assiste repetidas vezes, o efeito esmaece porque a surpresa visceral só acontece uma vez.

Isso também acontece com o uso do Apple Vision Pro. Uma vez você experimenta o headset de realidade expandida, a reação visceral é ficar boquiaberto com o poderoso eye-tracking e gestures e com os display com uma resolução nada modesta. Esse é o primeiro nível do design emocional, segundo Donald Norman: a reação visceral a aparência de um objeto.

Comportamental

Entretanto, uma vez você se habitua com o funcionamento intuitivo do Vision Pro, a experiência provida pelo hardware é atingida ao seu máximo e, para o fluxo de atenção ser mantido, se espera softwares tão poderosos quanto o headset para prolongar o efeito wow. E temos: jogos, experiências e apps que usufruem ao máximo dos sensores, alto-falantes e lentes do hardware.

Em uma avaliação geral, pode parecer que essa tecnologia ainda não tem espaço no mundo atual. Para alguns, pode parecer estranho usar um óculos que ilude sua percepção espacial diariamente. Porém, desacreditar no potencial da tecnologia e taxá-la como fracasso pode ser uma posição radical em um momento em que estamos construindo comportamento.

Em 2008, quando a App Store foi anunciada, parte das poucas aplicações disponíveis eram apps que usufruíam da câmera, giroscópio ou da tecnologia Multi-Touch da tela, ou seja, os desenvolvedores estavam experimentando as potencialidades enquanto os consumidores avaliavam a experiência de uso do iPhone. Eventualmente, as aplicações se tornaram mais úteis do que provocar risada com amigos com um simulador de chopp e, sucessivamente, sugeriram hábitos que só são possíveis devido a esses apps. No segundo nível, comportamental, há interação com o objeto, é o prazer e a efetividade do uso.

Reflexivo

Não é possível ter certeza como o Vision Pro irá se encaixar na rotina de seus usuários mas se cada iPhone, iPad e Mac tem funções ou características específicas que justificam seu uso, o headset também pode encontrar seu lugar. Enquanto nós desconhecemos sua potencialidade, não é possível limitar sua capacidade. Afinal, quando Henri Ford construiu o primeiro automóvel, é provável que os consumidores estavam esperando apenas um cavalo mais rápido. Jogar, projetar, trabalhar, assistir, aprender, relaxar... qualquer um desses verbos podem, em um futuro próximo, se relacionar com o uso do Vision Pro.

No terceiro estágio do design emocional, a experiência adquirida nos estágios anteriores permite a reflexão, a identificação, a satisfação pessoal e a formação de uma opinião ou um aprendizado. Para isso, é preciso tempo para perceber a reação coletiva acerca da tecnologia e dar tempo para ela ser apropriada não só por seus consumidores mas também pela própria equipe que o construiu e desenvolvedores que enriquecem e prolongam a experiência com o produto.

Referências

NORMAN, Donald A. *Design emocional: por que adoramos (ou detestamos) os objetos do dia-a-dia*. Rio de Janeiro: Rocco, 2008.

PAVON, Rafael. *Apple's Vision Pro allows us to dream big. Now it's time to make those blue sky experiences a reality*. CreativePool. 10 de março de 2024. Disponível em: <https://creativepool.com/magazine/features/apples-vision-pro-allows-us-to-dream-big--now-its-time-to-make-those-blue-sky-experiences-a-reality.30562>. Acesso em: 15 jan. 2025.



PROCESSO

FEEDBACK: ALGO DE ERRADO NÃO ESTÁ CERTO

TEXTO

CAROL PARENTE

ILUSTRAÇÃO

MARCELLE RIBEIRO

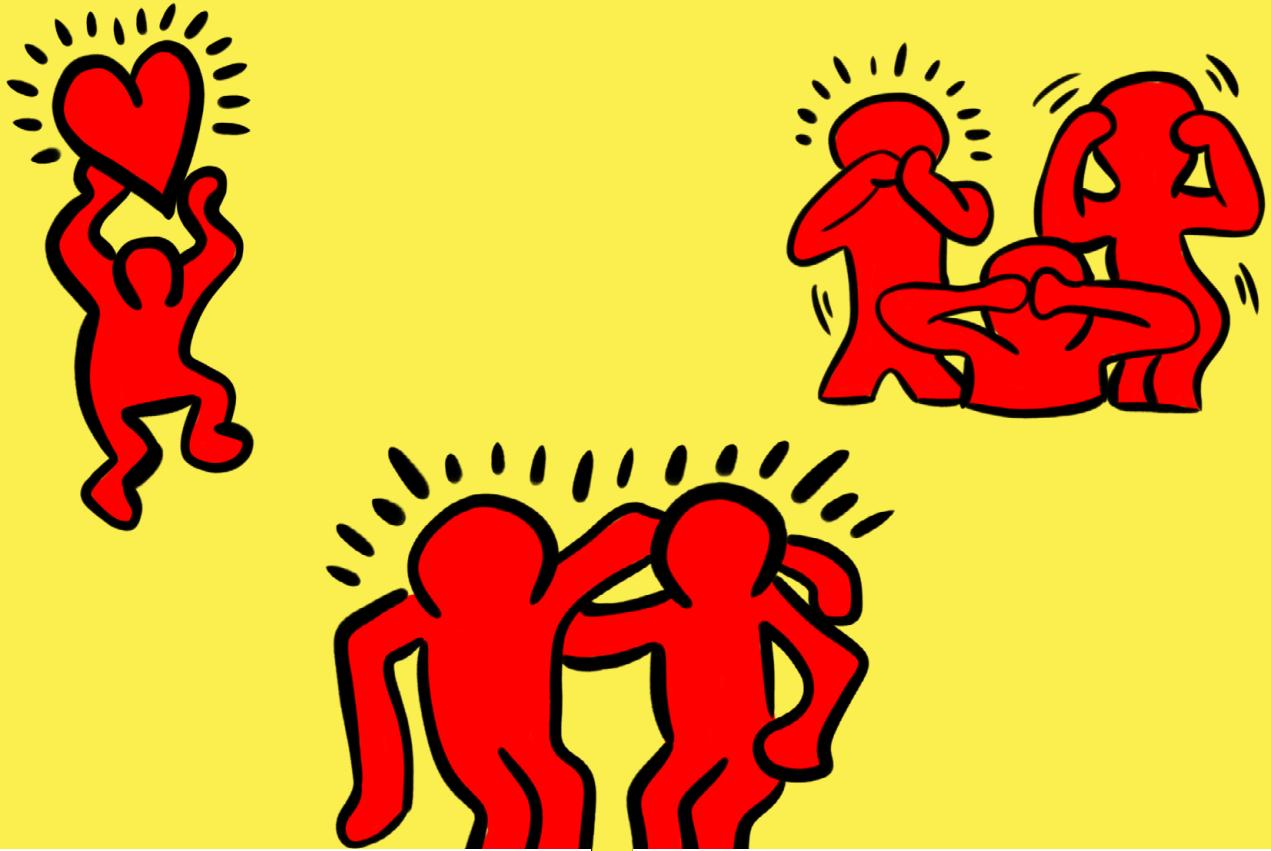
REVISÃO

LUCAS BAPTISTA

O feedback, uma palavra que remete a tantas reações diferentes, pode ser ao mesmo tempo construtivo e destrutivo, esclarecedor e confuso. Nas organizações e nas relações pessoais, ele surge como uma ferramenta fundamental para o desenvolvimento, mas nem sempre é recebido da maneira como foi planejado. O título "Algo de errado não está certo" reflete justamente esse paradoxo: por que, muitas vezes, feedbacks que têm a intenção de ser úteis são recebidos com resistência, dor ou incompreensão?

Neste texto, vamos explorar a complexidade do feedback, entender por que ele pode desencadear reações tão diversas e como é possível, em vez de se machucar, crescer a partir dele. Baseado nas ideias de Douglas Stone e Sheila Heen no livro **"Obrigado pelo feedback"**, vamos desmembrar a anatomia de uma conversa difícil e aprender a navegar pelas situações mais espinhosas.





Feedbacks estão em todo canto

Vivemos em um mundo onde feedbacks estão em toda parte. No trabalho, são entregues formalmente em avaliações de desempenho ou de maneira mais casual em reuniões de equipe. Nas relações pessoais, o feedback aparece nas interações cotidianas, seja entre amigos, familiares ou parceiros românticos. Até mesmo nas redes sociais, o feedback é onipresente, manifestando-se em curtidas, comentários e compartilhamentos.

Mesmo quando não solicitamos, o feedback nos alcança. Somos constantemente expostos a julgamentos sobre nossas ações, opiniões e comportamentos, e é inegável que essa exposição nos afeta de maneiras profundas. Por mais que muitos de nós desejemos crescer, o processo de receber feedback frequentemente nos coloca em uma posição defensiva. Afinal, ninguém gosta de se sentir criticado, mesmo quando o objetivo é nos ajudar.

A anatomia de uma conversa difícil

Não é surpresa que algumas das conversas mais difíceis da nossa vida sejam sobre feedback. Quando ouvimos algo que desafia nossa maneira de ser, agir ou pensar, uma variedade de reações emocionais pode surgir. Algumas dessas conversas são marcadas por desconforto, tensão e, em alguns casos, até ressentimento.

Douglas Stone e Sheila Heen apresentam em seu livro a ideia de que essas conversas difíceis geralmente envolvem três tipos de "gatilhos de feedback" que explicam por que certos retornos são tão dolorosos ou difíceis de aceitar. Mas antes de explorarmos esses gatilhos, é importante reconhecer que muitas dessas conversas desafiadoras começam com a nossa própria predisposição de receber o feedback. Cada pessoa carrega uma "identidade" única, composta por crenças, experiências e emoções, que influenciam diretamente como processamos e reagimos ao que ouvimos dos outros.



Gatilhos de feedback

Os gatilhos de feedback são mecanismos internos que influenciam nossa reação imediata a críticas, sugestões ou elogios. Esses gatilhos são responsáveis por acionar as defesas naturais do ser humano, e cada um deles oferece uma pista sobre por que certos feedbacks são mais difíceis de digerir do que outros.

Gatilho Identidade

O Gatilho Identidade está diretamente ligado a quem somos e como nos vemos. Quando recebemos feedbacks que parecem ameaçar nossa visão de nós mesmos, entramos em um estado de alerta. Nossa identidade é fundamental para nosso senso de autoestima, e qualquer comentário que questiona nossas habilidades, moralidade ou caráter pode ser percebido como um ataque pessoal.

Por exemplo, imagine uma pessoa que sempre se orgulhou de ser muito organizada e produtiva, mas recebe um feedback do seu chefe sugerindo que sua gestão de tempo precisa melhorar. Esse feedback pode desencadear uma crise interna, pois a sugestão de que algo está errado contradiz a identidade que essa pessoa construiu ao longo da vida. A reação imediata pode ser negar ou desqualificar o feedback para proteger sua autoestima.

Gatilho Verdade

O Gatilho Verdade é acionado quando questionamos a veracidade ou precisão do feedback que recebemos. Muitas vezes, nossa primeira reação é rejeitar o feedback porque acreditamos que ele está incorreto ou desinformado. Este gatilho nos desafia a questionar a validade do que ouvimos: "Será que isso é realmente verdade? Esse comentário faz sentido? Essa pessoa entende a situação por completo?"

Imagine que um colega de trabalho sugira que você não está colaborando o suficiente em um projeto de equipe. Você pode sentir imediatamente que essa percepção está errada, talvez porque acredita que está fazendo o máximo possível com os recursos que tem. Nesse caso, o feedback pode ser descartado antes mesmo de ser considerado porque o gatilho da verdade foi ativado.

Gatilho Relacionamento

O Gatilho Relacionamento é influenciado pela dinâmica entre você e a pessoa que está dando o feedback. Às vezes, não é o conteúdo do feedback que nos perturba, mas quem o está entregando. Se temos uma relação complicada com essa pessoa, ou se não confiamos nela, podemos descartar o feedback simplesmente porque vem de alguém que não respeitamos ou gostamos.

Esse gatilho também pode funcionar ao contrário: quando alguém de quem gostamos muito nos dá feedback, podemos aceitar sem questionar, mesmo que o comentário não seja construtivo ou bem informado. Nossas reações ao feedback, portanto, estão profundamente ligadas às emoções e percepções que temos sobre quem o está oferecendo.

Crescer em vez de se machucar

Receber feedback é inevitável, mas como reagimos a ele é algo que podemos controlar e aprimorar. Em vez de permitir que os gatilhos nos machuquem, podemos aprender a crescer a partir deles. Aqui estão algumas dicas práticas para lidar com cada gatilho:

Lidando com o Gatilho Identidade

Para mitigar os efeitos do Gatilho Identidade, é importante separar quem você é daquilo que você faz. Feedback sobre seu comportamento ou desempenho não define o seu valor como pessoa. Um bom exercício é praticar a auto-empatia e lembrar que todos cometemos erros e temos áreas para melhorar. Tente observar o feedback como uma oportunidade de auto-conhecimento e desenvolvimento, em vez de uma ameaça à sua identidade.



Lidando com o Gatilho Verdade

Com o Gatilho Verdade, a chave é a curiosidade. Em vez de descartar o feedback imediatamente, faça perguntas. Tente entender de onde a outra pessoa está vindo e quais são as evidências por trás de suas palavras. Você pode não concordar totalmente com o feedback, mas pode haver uma verdade parcial que merece sua atenção. Procure pelo menos um aspecto no qual você pode crescer, mesmo que o feedback não seja 100% preciso.

Lidando com o Gatilho Relacionamento

Ao lidar com o Gatilho Relacionamento, tente focar no conteúdo do feedback, não na pessoa que o está oferecendo. Pergunte a si mesmo: "Se outra pessoa tivesse dito isso, eu reagiria da mesma forma?" Muitas vezes, nossas emoções sobre a pessoa distorcem nossa capacidade de ouvir o que está sendo dito. Manter uma postura de escuta ativa e objetiva pode ajudar a neutralizar o impacto do relacionamento na qualidade do feedback.



O feedback é uma das ferramentas mais poderosas para o crescimento pessoal e profissional, mas também uma das mais desafiadoras. Ao compreender os gatilhos que afetam nossa recepção de feedback e aprender a abordá-los de maneira construtiva, podemos transformar conversas difíceis em oportunidades de aprendizagem. Afinal, o que parece errado à primeira vista pode ser justamente o que nos leva a fazer algo certo no futuro.

DESIGN

10 HEURÍSTICAS DE NIELSEN (TAYLOR'S VERSION)

TEXTO

MARIA EDUARDA

ILUSTRAÇÃO

GILBERTO NETO

REVISÃO

LUCAS BAPTISTA



Recentemente, a cantora Taylor Swift percorreu diversos países no mundo em sua turnê "The Eras Tour" que, ao longo das quase três horas de show, promove canções de todos seus álbuns. Nessa turnê, ela demonstrou uma conexão poderosa com os fãs, criando uma experiência memorável por meio de inúmeras referências, uma estrutura projetada para proporcionar vivências únicas e uma performance digna de um espetáculo teatral. Além disso, Taylor transmitiu uma forte sensação de pertencimento, mesmo para aqueles que a acompanhavam apenas por meio de uma tela.

Essa experiência me fez recordar um conceito que estudei em Interação Humano-Computador, que oferece ferramentas e metodologias para desenvolver interfaces voltadas ao usuário: as 10 heurísticas de Nielsen. Com o avanço da tecnologia, tornou-se necessário projetar designs que facilitem a usabilidade, tornando aplicativos, sistemas operacionais e plataformas cada vez mais fáceis de serem compreendidos e utilizados. Para atender a essa demanda por praticidade, flexibilidade e facilidade, Jakob Nielsen, pesquisador de Interação Humano-Computador e cofundador da Nielsen Norman Group, criou, em 1994, as 10 heurísticas de usabilidade para interfaces de usuário. Nesse artigo, iremos revisar as heurísticas propostas por Nielsen a partir de estratégias usadas por Taylor Swift para promoção de seus álbuns.

Visibilidade do Status do Sistema – O Álbum “Midnights” e a Estratégia de Lançamento

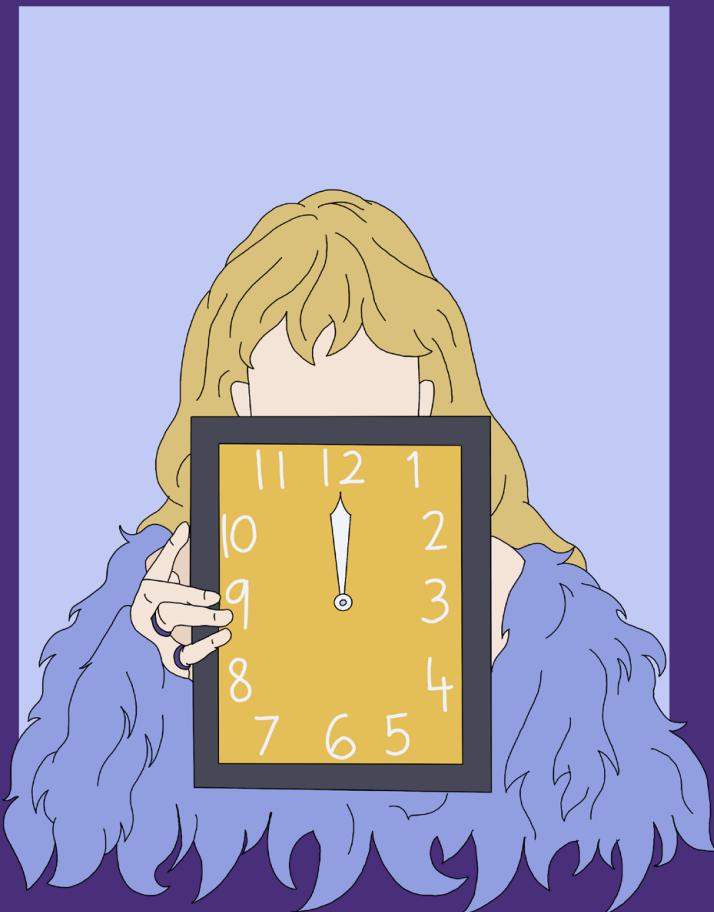
No lançamento do álbum *Midnights* (2023)*, ela conduziu seu público por cada etapa do processo, com teasers, anúncios de faixas, contagens regressivas, e até referências sutis em suas roupas. Essa visibilidade é essencial em interfaces: o usuário precisa sempre saber o que está acontecendo.

É fundamental incluir elementos que mostrem o progresso ou o que pode ocorrer após uma ação em produtos digitais. Isso evita a incerteza sobre a conclusão de uma ação, como enviar uma mensagem ou apertar um botão, e proporciona uma experiência mais segura e informada. Recursos como indicadores de progresso, barras de carregamentos e notificações de confirmação auxiliam a visibilidade do status.

Correspondência entre o Sistema e o Mundo Real – O Álbum “Speak Now” e a Música “Long Live”

A interação de Taylor Swift com seus fãs reflete uma linguagem comum que ela cultiva, especialmente em músicas como *Long Live*, onde celebra essa conexão e aproxima o seu mundo do mundo deles.

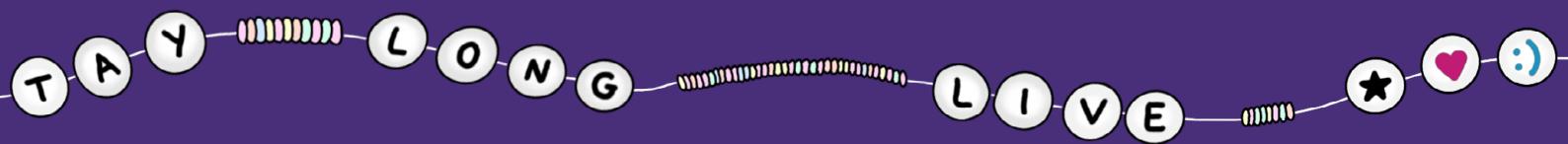
Em interfaces, essa conexão também é essencial: considerar o conhecimento prévio e expectativas das pessoas e usar uma linguagem comum e elementos reconhecíveis cria uma relação mais natural entre o sistema e o usuário. Para isso, utilizar linguagem comum ao usuário, utilizar metáforas e referências do mundo real e utilizar storytelling para construir interfaces reconhecíveis proporcionam uma experiência de uso segura e envolvente.



Controle e Liberdade do Usuário – “Midnights” e as Pulseiras de Amizade

Na The Eras Tour, os fãs criaram um movimento único ao trocarem pulseiras de amizade nos shows, inspirados por uma citação em uma de suas músicas que fazia referência a essa troca. Esse gesto de liberdade e personalização, adotado em todos os continentes por onde a turnê passou, reflete como os fãs podem participar de uma experiência simbólica e autêntica. Em um sistema, é essencial oferecer aos usuários essa mesma sensação de autonomia e controle nas interações.

Ao utilizar menus bem estruturados e organizados, botões de “desfazer”, cancelar ou sair e permitir personalização de layouts e configurações permitem que os usuários ajustem as experiências conforme preferirem.



Consistência e Padronização – O Número 13

Ao longo de sua carreira, Taylor Swift fez do número 13 sua marca registrada, tornando-o um símbolo que seus fãs reconhecem imediatamente. Desde a turnê de seu segundo álbum, Taylor usou o número 13 como um sinal que permeia tudo, desde a duração de músicas até a quantidade de faixas. Da mesma forma, uma interface deve manter um padrão visual e funcional consistente para criar uma experiência de familiaridade para os usuários.

Isso evita que o usuário precise reprender conceitos ou regras de interação, tornando a experiência mais intuitiva e agradável, assim como Taylor faz ao reforçar suas referências para seus fãs. Para implementar essa consistência, é preciso manter uma estrutura hierárquica lógica e um posicionamento coerente dos botões em todo o sistema. Além disso, é preciso utilizar termos claros e sem ambiguidades, evitando jargões.

Prevenção de Erros – “Folklore” e a “Long Pond Studio Session”

Ao lançar **Folklore**, Taylor Swift preveniu “erros” artísticos ao se afastar das expectativas de um estilo extravagante, trazendo um álbum intimista e centrado em histórias de terceiros. Da mesma forma, ao criar interfaces, é fundamental planejar a prevenção de erros de maneira a proporcionar uma experiência satisfatória e sem frustrações.

A quinta heurística de Nielsen se aplica diretamente aqui, pois desenvolvedores de produtos digitais devem antecipar e evitar erros que possam ocorrer durante a interação do usuário com o sistema. Por isso, validar a entrada de dados com feedbacks em tempo real e promover testes de usabilidade são ações essenciais para garantir a confiança e satisfação ao usar uma interface.



Reconhecimento em vez de Memória – A “Faixa 5”

Desde o início de sua carreira, Taylor Swift criou padrões reconhecíveis, especialmente usando números. Um dos exemplos mais queridos pelos fãs é a “faixa 5”: em todos os seus álbuns, ela tem a tradição de ser a faixa mais melancólica, vulnerável e pessoal, refletindo o estado emocional de Taylor ao escrever aquele álbum. Da mesma forma, para uma experiência de usuário agradável, Jakob Nielsen sugere que o sistema deva ser reconhecível em vez de exigir que o usuário memorize novos atalhos, terminologias ou fluxos. Ao utilizar rótulos claros e descritivos e reaproveitar padrões de interação e elementos visuais, facilita a interação e reduz a carga de memória do usuário.

Eficiência e Flexibilidade de Uso – Álbum “1989” e o Single “Shake It Off”

Taylor Swift transformou sua carreira ao lançar **1989**, seu álbum de estreia no mundo pop, ampliando seu público ao adotar uma nova estética e sonoridade mais acessível, com sucessos como “Shake It Off”. Ao entender o que ressoaria com o público em geral, Taylor construiu uma ponte entre seu estilo único e a familiaridade do pop, criando um sucesso que conquistou fãs novos e antigos. Da mesma forma, uma interface eficaz e flexível permite que usuários novos e avançados encontrem rapidamente suas interações ideais, adaptando-se às necessidades de cada um.

Eficiência em uma interface significa que o sistema facilita a realização de tarefas de forma rápida e simples, com fluxos de trabalho otimizados. Proporcionar eficiência e flexibilidade ajuda usuários de diferentes níveis de experiência a interagir de maneira intuitiva e personalizada. Para isso, dar opções de personalização e oferecer funcionalidades avançadas que não sejam fundamentais para o uso principal da interface.

Estética e Design Minimalista – “Evermore” e a Estética Teatral

No álbum **Evermore**, Taylor Swift optou por uma abordagem simples e discreta, entendendo que ele pedia uma estética menos chamaativa, com tons terrosos e minimalismo visual.

Em interfaces, o design minimalista age de forma semelhante, eliminando informações desnecessárias e focando nos elementos essenciais para facilitar a interação. Ao aplicar o minimalismo reduzindo o excesso de elementos e escolhendo uma paleta de cores harmoniosa, a experiência fica visualmente agradável e intuitiva.

Recuperação diante de erros – Álbum Reputation

Depois de uma fase turbulenta, Taylor Swift lançou **Reputation** como resposta às controvérsias e críticas recebidas, mostrando uma recuperação forte e estratégica de sua imagem. Esse processo de superação e adaptação também é essencial para interfaces: permitir que o usuário se recupere facilmente de erros e siga em frente na experiência de uso.

Sistemas devem fornecer ao usuário feedbacks descritivos e compreensíveis quando algo dá errado. Também é necessário que o sistema ofereça passos para resolver o problema. Entretanto, para evitar frustrações, mostre alternativas ou oriente o usuário antes que ele cometa um erro comum.

Ajuda e documentação – Álbum de estreia Taylor Swift

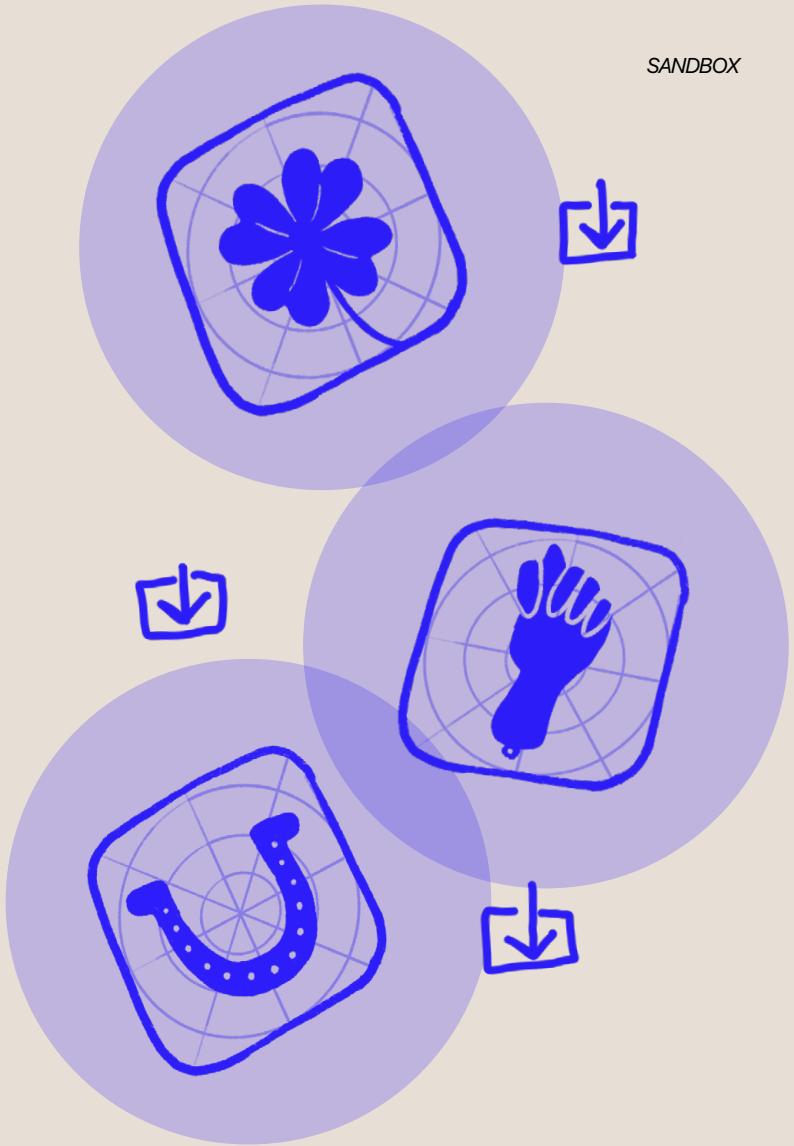
Desde seu álbum de estreia, Taylor Swift sempre se esforçou para fornecer contexto e histórias para seus fãs através de suas letras. Ela compartilha sua perspectiva e experiências de vida, criando uma conexão emocional com seu público. Da mesma forma, em interfaces, a ajuda e a documentação são fundamentais para que os usuários entendam como navegar pelo sistema e solucionar problemas que possam surgir.

Para garantir que os usuários tenham acesso a informações úteis e relevantes, é importante criar materiais de apoio que sejam claros, acessíveis e compreensíveis. Disponibilizar uma seção de perguntas frequentes, tutoriais, guias e um canal de feedbacks dos usuários auxiliam a navegação e resolução de problemas.

PROCESSO

ALÉM DOS ZEROS E UNS: ENTENDENDO AS MÉTRICAS DO MUNDO DIGITAL

Existem coisas na vida que desafiam a lógica. A famosa sorte de principiante, a "lei do ex" no futebol, o número da sorte ou do azar, a simpatia que promete trazer de volta o amor em 3 dias... Essas peculiaridades fazem parte do nosso cotidiano: há quem deboche e há quem acredite. Convenhamos: na nossa vida, é quase impossível não acreditar em algo inexplicável. Mas, em um mundo feito de zeros e uns, nada disso tem vez.

*TEXTO***FELIPE FRANCES***ILUSTRAÇÃO***RAISSA PARENTE***REVISÃO***LUCAS BAPTISTA**

No mundo digital, os números ditam as regras. E isso vai além do binário, programador! Não basta ser bom em código. Hoje, onde apps competem por atenção em meio a um mar de ofertas digitais, o entendimento de métricas pode ser a ferramenta mais poderosa para desenvolvedores independentes que buscam não apenas sobreviver, mas prosperar. Enquanto grandes empresas investem milhões em dados, o desenvolvedor independente tem nas métricas o mapa para extrair insights valiosos e maximizar o retorno de cada linha de código.

Por que Métricas Importam no Mundo Digital?

Em um mercado onde todos os dias surgem novos aplicativos, a concorrência pela atenção do usuário é **BRUTAL**. Com tantas opções disponíveis, os desenvolvedores independentes enfrentam o desafio de destacar seus apps sem o respaldo financeiro e de dados que grandes empresas possuem. Nesse contexto, contar apenas com a intuição ou com o talento para programação não é suficiente. É aí que entram as métricas – ferramentas que revelam exatamente como os usuários interagem com seu app, o que mais os atrai e onde eles perdem o interesse.

Ao monitorar as métricas, você ganha acesso a insights práticos que ajudam a tomar decisões fundamentadas. Em vez de depender de sorte ou de suposições, você passa a entender o que funciona e o que precisa de ajustes. Isso significa que cada atualização ou modificação no app pode ser orientada por dados reais, maximizando as chances de sucesso. É o mesmo princípio que grandes empresas usam para crescer: dados ajudam a ajustar estratégias, a otimizar a experiência do usuário e a direcionar o crescimento do produto.

Mas não pense que essa abordagem é exclusiva das corporações. Como desenvolvedor independente, você também pode – e deve – adotar uma mentalidade orientada por dados.

Com as métricas certas, até o menor dos aplicativos pode se destacar ao entender o que realmente importa para o seu público. E, ao aplicar esses insights, você passa a ter uma vantagem competitiva poderosa: a capacidade de evoluir constantemente e de alinhar seu app às expectativas e necessidades dos usuários.

Assim, com a base sólida das métricas, o próximo passo é saber **quais números realmente importam**. Vamos agora às métricas fundamentais que todo desenvolvedor precisa entender para transformar dados em resultados.



TRAGO
SEU APP
P/ LOUAR
EM
3 DIAS

AS MÉTRICAS FUNDAMENTAIS PARA DESENVOLVEDORES DE APPS

No mundo digital, as métricas não são apenas números: são sinais claros de como seu aplicativo está performando e onde ele precisa de ajustes. Entender as métricas certas é o que permite otimizar o app com precisão, mesmo com recursos limitados. Vamos explorar as métricas essenciais e entender como cada uma delas pode ser uma peça-chave no seu plano de crescimento:



Impressões

Imagine que o seu app é uma loja e que a App Store seja o shopping. O número de impressões é quantas vezes as pessoas que estão no shopping passam na frente da sua loja, mesmo sem entrar.

Em outras palavras, as impressões representam a quantidade de vezes que o ícone do app aparece para os usuários na App Store, seja nos resultados de busca, nas seções de destaque ou em listas. Essa métrica revela a visibilidade inicial do app.



Visualizações da Página do Produto

De todas as pessoas que passaram em frente à sua loja, algumas resolveram entrar para conhecer. Esta é a métrica de visualizações.

Ela mostra quantas pessoas visitaram a página do seu app, indicando o interesse em saber mais sobre ele e reflete o quanto atraente é a apresentação inicial do app na loja.



Taxa de Conversão

São os clientes que entraram na sua loja do shopping e compraram o seu produto. Ou no caso do seu aplicativo, os usuários que fizeram o download dele.

A taxa de conversão é a relação entre o número de visualizações da página do produto e o número de downloads. Em resumo, ela mostra a eficácia da página do app em transformar visitantes em usuários.

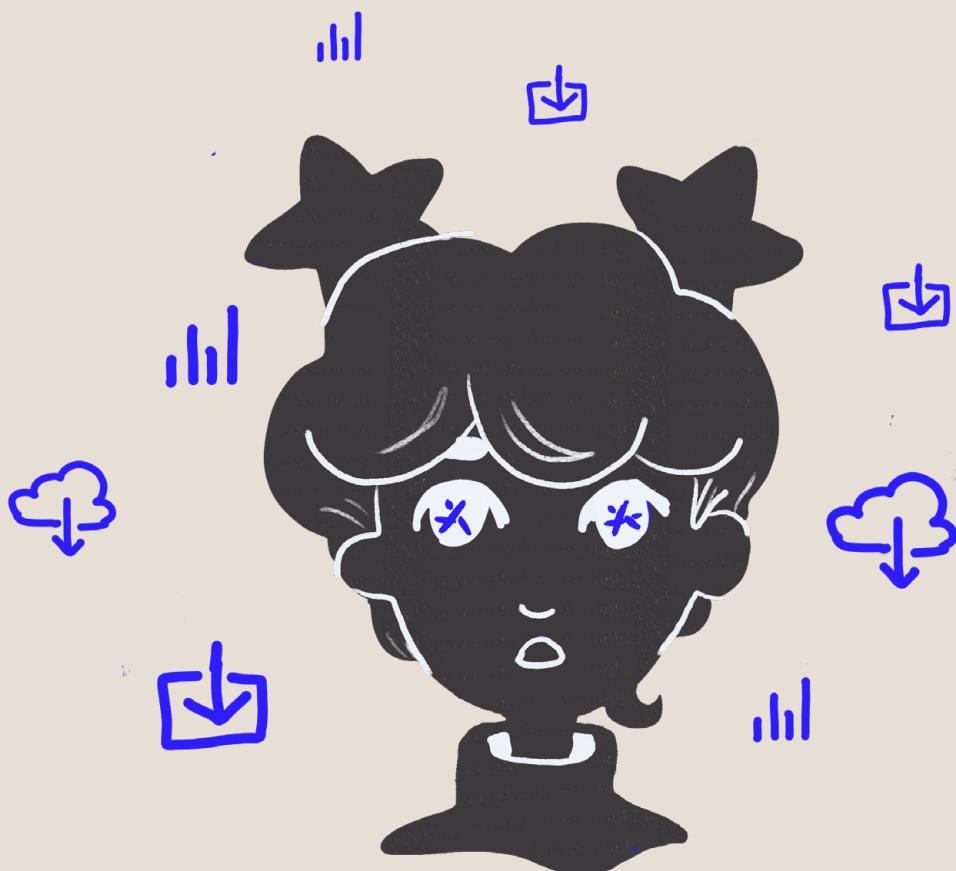
Downloads por data

Repete a analogia do shopping todos os dias, semanas, ou meses.

Os downloads iniciais indicam a aceitação do app, enquanto a manutenção da taxa diária ou semanal, por exemplo, são um termômetro da popularidade e da retenção do seu app.

COMO AS MÉTRICAS PODEM GUIAR SUAS DECISÕES ESTRATÉGICAS?

Uma vez que você comprehende as métricas fundamentais, o próximo passo é **usá-las como um guia para melhorias estratégicas no app**. Cada métrica oferece um insight valioso, mas é na análise combinada dessas informações que você encontra o caminho para fazer ajustes que impactem diretamente o sucesso do aplicativo. Aqui estão maneiras de interpretar essas métricas e te ajudar na sua tomada de decisões:



Muitas impressões, poucas visitas

Se as impressões do seu app são altas, mas poucas pessoas estão visitando a página do produto, isso indica que, embora o app esteja visível, ele não está chamando atenção o suficiente para que as pessoas queiram saber mais. Uma análise simples pode ajudar a identificar o problema: o ícone está comunicando bem a proposta do app? O nome é claro e atraente?

Muitas visitas, poucos downloads.

Se as visualizações da página do produto são satisfatórias, mas a taxa de conversão para download é baixa, isso sugere que os usuários estão interessados em saber mais, mas não estão convencidos a baixar o app. Isso pode acontecer por uma série de elementos da página. Esse é o momento de reavaliar o conteúdo da página do produto.

Número de downloads diminuindo com o tempo.

Se os downloads iniciais foram bons, mas o crescimento desacelerou, é provável que o app esteja precisando de novidades para manter os usuários engajados ou um pouco de marketing.

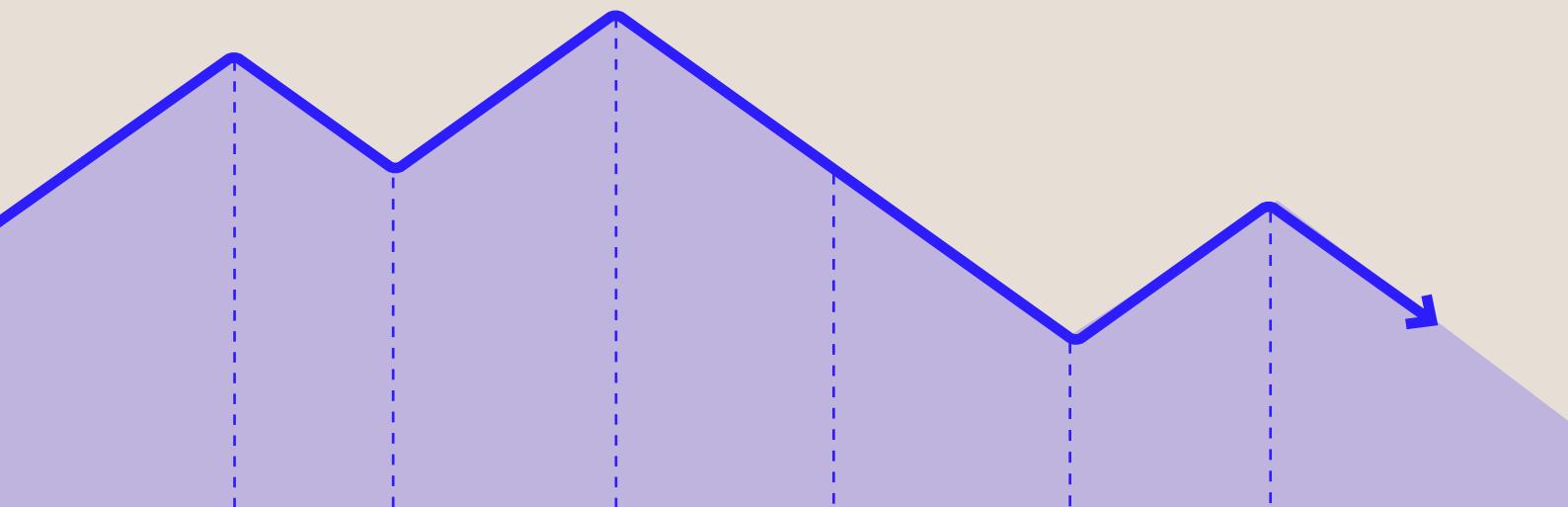
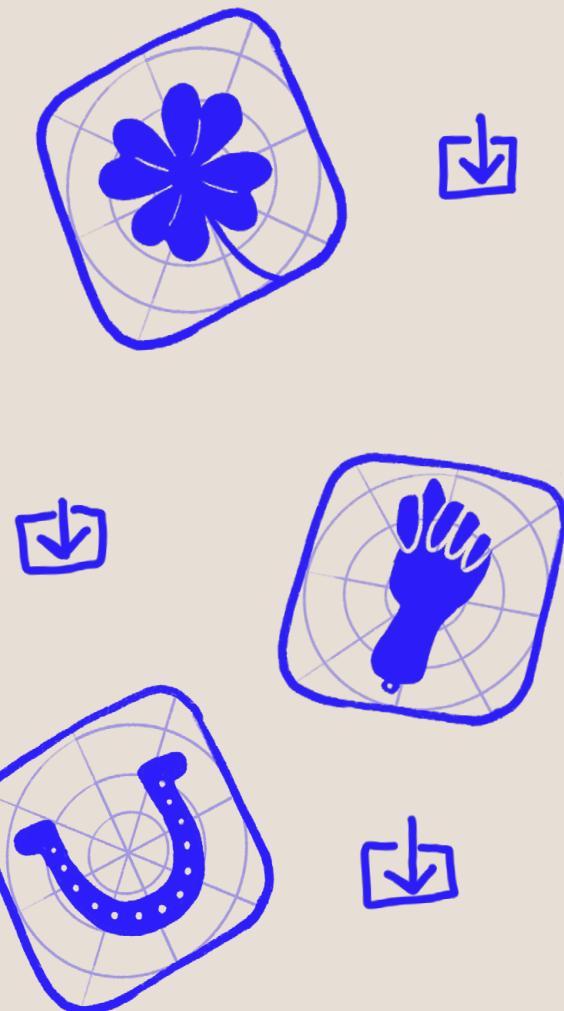
Além disso, se poucos usuários estão atualizando o app, isso pode ser um sinal de que as atualizações não são percebidas como relevantes.

Como você pode perceber, cada métrica isolada já oferece uma visão importante, mas é ao combiná-las que você obtém uma análise mais profunda.

Ao analisar as métricas estrategicamente, você transforma dados em decisões que podem impulsionar o crescimento do app.

Como desenvolvedor, você tem o poder de adaptar cada detalhe do seu app com base nos dados coletados, ajustando sua estratégia para alinhar o produto às expectativas e necessidades do usuário. Portanto, comece a monitorar suas métricas!

O sucesso do seu aplicativo não depende de sorte nem simpatia, e sim de números. E números que vão além dos zeros e uns.





PROCESSO

COMO OS X-MEN UTILIZAM OS PRINCÍPIOS DE FORTES EQUIPES DE PRODUTO

TEXTO

JOÃO ROBERTO

ILUSTRAÇÃO

DANIEL OPPELT

REVISÃO

PAULO HENRIQUE MELO

ELIS VIEIRA

Para quem cresceu nos anos 2000 a 2010, assistir a "X-Men: Evolution" é um clássico que já lembra o clima do almoço em casa. Mas a série animada é apenas uma das várias produções que ajudaram a tornar os X-Men um fenômeno mundial, com repercussão em séries de TV, filmes, games, brinquedos e quadrinhos.

Embora tenha começado como uma história em quadrinhos de ficção científica, a franquia X-Men é marcada por temas profundos e universais, como preconceito, política e família. Em um mundo onde a humanidade evoluiu e alguns indivíduos adquiriram o gene X, ganhando habilidades além do normal, surgem conflitos entre os mutantes e o restante da sociedade. Como resposta, o Professor Charles Xavier funda uma escola que acolhe esses jovens mutantes, ensinando-os a dominar seus poderes e, ao mesmo tempo, defendendo o sonho de uma convivência pacífica entre humanos e mutantes.

Com essa realidade apresentada, podemos extrair diversas lições valiosas dos X-Men para a construção de equipes de produto excepcionais. Venha comigo descobrir quais princípios de fortes times de produto regem esse time de mutantes.

Equipe de missionários, não mercenários

Os X-Men são um grupo de missionários, movidos por uma causa maior: a defesa dos direitos dos mutantes e a coexistência pacífica entre eles e os humanos. Eles não são motivados por recompensas materiais, mas por uma visão clara e uma causa compartilhada.

Esse princípio se aplica a equipes de produto quando elas estão profundamente conectadas com o propósito do produto. Em vez de serem apenas executores de tarefas (mercenários), uma equipe missionária está motivada por uma missão clara. Eles não estão ali apenas para "fazer o trabalho", mas porque acreditam no impacto que o produto pode ter na vida dos usuários e no mercado, além de que são comprometidos em resolver problemas de seus clientes. Essa conexão com a missão resulta em maior envolvimento, comprometimento e uma busca constante por soluções inovadoras.

Empoderamento e responsabilidade da equipe

Nos X-Men, cada membro da equipe é empoderado para usar suas habilidades únicas. O Professor Xavier dá a seus mutantes a liberdade de tomar decisões no campo de batalha, confiando em suas capacidades e julgamento. Esse empoderamento vem com uma grande responsabilidade: cada X-Men é responsável não só por suas próprias ações, mas pelo bem-estar de toda a equipe e pelo sucesso da missão.

Equipes de produto de sucesso funcionam de maneira semelhante. Cada membro da equipe é empoderado para resolver problemas difíceis para o negócio dentro de sua área de expertise. Esse empoderamento, no entanto, é acompanhado pela responsabilidade de entregar resultados e pelo compromisso com a qualidade do produto, muitas vezes utilizando OKRs (Objective Key Results).

Estrutura hierárquica do time

Nos X-Men, existe uma estrutura clara de liderança, com o Professor Xavier no topo. Entretanto, dentro das missões, a hierarquia não é tão presente. Ciclope pode ser o líder em campo, mas ele confia na experiência dos outros membros, como Tempestade ou Wolverine, quando necessário. A estrutura não é rígida e muda conforme a situação demanda.

Nas equipes de produto, a hierarquia deve ser inexistente, todos são colaboradores individuais e não há um gerente de pessoas. Embora o gerente de produto tenha responsabilidade sobre certas áreas, é importante ressaltar que ele não é chefe de ninguém na equipe de produto. A estrutura do time deve

ser mais colaborativa do que rígida, permitindo que o melhor talento lidere de acordo com as necessidades do projeto e do momento. Naturalmente, no caso de reportar problemas, as pessoas conversam com seus respectivos gerentes funcionais, um engenheiro reporta ao head de engenharia, tão quanto os designers e os profissionais de produto também reportam aos seus respectivos heads.



Colaboração do time

A colaboração é central para o sucesso dos X-Men. Mesmo com habilidades únicas, os mutantes só conseguem alcançar suas metas quando trabalham juntos. Cada missão exige que eles combinem seus poderes para superar desafios e derrotar inimigos poderosos. Nenhum X-Man é mais importante que o outro, e a chave do sucesso está em como eles colaboram de forma eficaz.

Da mesma forma, equipes de produto devem colaborar intensamente. A criação de um produto exige integração entre designers, engenheiros, gerentes de produto e outras funções. O trabalho em equipe é o que possibilita a construção de algo maior do que a soma das partes. A capacidade de trabalhar juntos, resolver problemas e se comunicar é o que faz com que o produto evolua continuamente.

Escopo da equipe

Assim como nos X-Men, em que cada mutante possui uma habilidade única e complementa a equipe como um todo, as equipes de produto precisam dividir suas responsabilidades de forma inteligente, mas coordenada. O Professor Xavier e os líderes dos X-Men, assim como os heads de uma empresa, devem equilibrar habilidades e forças dentro de uma equipe maior para que todos trabalhem em sinergia, focados em uma missão comum.

É fundamental que uma equipe de produto assume a responsabilidade por toda a experiência do produto, mas é comum, especialmente em empresas maiores, que cada equipe seja responsável por uma parte específica da experiência do produto. Essa divisão é relativamente simples em startups menores, ou times de challenge do Academy. No entanto, em uma empresa maior, a situação se complica, e muitas vezes os times acabam sendo separados com base em diferentes personas, dispositivos, ou, mais frequentemente, pela arquitetura do software. Por isso, o alinhamento entre o head de produto e o head de engenharia é essencial para definir o tamanho e o escopo das equipes. Não existe uma fórmula perfeita para dividir as tarefas: ao otimizar uma área, outra pode ficar em segundo plano, o que torna a priorização um elemento chave.

Estabilidade da equipe

Os X-Men, apesar das mudanças e adições ao longo do tempo, têm uma formação básica que permanece estável. Essa estabilidade contribui para o entendimento profundo entre os membros, permitindo que antecipem os movimentos uns dos outros e construam uma sinergia sólida. Quando a equipe é estável, o trabalho flui de maneira mais eficiente e com menos fricções.

Equipes de produto que mantêm estabilidade ao longo de diferentes projetos tendem a ser mais eficazes. Quando os membros da equipe se conhecem bem e entendem as habilidades uns dos outros, conseguem colaborar de maneira mais fluida e resolver problemas de forma mais rápida. Trocas frequentes de membros podem quebrar esse ritmo, dificultando até a paixão como missionário, enquanto uma equipe estável cria uma base sólida para inovação e crescimento contínuo. É quase impossível ter um time de missionários quando o projeto dura curtos espaços de tempo e depois se desmembra, o que justifica o fato de que muitos apps visionários do Academy não se manterem após os challenges.

Autonomia da equipe

Dentro dos X-Men, cada personagem tem autonomia para agir conforme sua especialidade. Quando estão em missão, não esperam por aprovações para tomar decisões críticas em campo. Essa autonomia é uma marca registrada do time, que confia nas habilidades e julgamento dos membros para tomar as melhores decisões em tempo real.

Equipes de produto altamente eficazes precisam de um nível semelhante de autonomia. Isso significa que a equipe pode decidir como executar suas tarefas da forma que acharem mais conveniente. Equipes autônomas conseguem ser mais ágeis, inovadoras e eficientes, pois têm liberdade para tomar decisões rápidas e iterar no desenvolvimento do produto, pois essa autonomia faz o times se sentir empoderado e cria uma paixão semelhante a de um missionário.

Em resumo, as lições dos X-Men mostram que uma equipe de produto excepcional é construída com base em propósito, autonomia e colaboração. Assim como nos mutantes de Xavier, uma equipe de produto bem-sucedida deve ter uma missão clara, valorizando a conexão entre seus membros e a causa que compartilham. O empoderamento, junto a uma estrutura hierárquica flexível, permite que cada pessoa contribua conforme sua especialidade, levando à inovação e eficácia.

Além disso, a estabilidade e a comunicação eficiente, mesmo com equipes distribuídas, fortalecem a conexão e a capacidade de antecipação entre os membros, favorecendo a evolução contínua do produto. Quando cada indivíduo entende seu papel e tem autonomia para agir, a equipe se torna mais ágil e criativa, pronta para enfrentar desafios complexos e entregar produtos impactantes. Os X-Men, em sua jornada, exemplificam o poder de equipes comprometidas, e seus princípios são guias valiosos para aqueles que buscam transformar suas equipes de produto em times verdadeiramente extraordinários.



CÓDIGO

EXPLORANDO A LINGUÍSTICA COMPUTACIONAL COM O FRAMEWORK NATURAL LANGUAGE

TEXTO

PAULO HENRIQUE MELO

ILUSTRAÇÃO

DAVID AUGUSTO

REVISÃO

DAVID AUGUSTO

GABRIELA BEZERRA



Entre palavras e algoritmos

A escrita e a tecnologia sempre foram áreas pelas quais tive interesse, sobretudo agora, como graduando em Engenharia de Computação e em Letras. Por um certo tempo, estudei as duas de forma separada, mas sempre busquei modos de conectá-las. "Como um compilador verifica se um código está escrito corretamente?"

"SERÁ QUE ELE VERIFICA A ESTRUTURA SINTÁTICA DESSE CÓDIGO?"

"A SEMÂNTICA DAS EXPRESSÕES IMPORTA NESSE CASO?"

"QUAL A RELAÇÃO EXISTENTE ENTRE A LINGUAGEM HUMANA E AS LINGUAGENS DE PROGRAMAÇÃO?"

Esses eram alguns dos paralelos que eu fazia quanto a elas, ainda sem saber da existência de um vasto campo capaz de promover a **interseção** entre as áreas de Letras e Ciência da Computação: a Linguística Computacional.

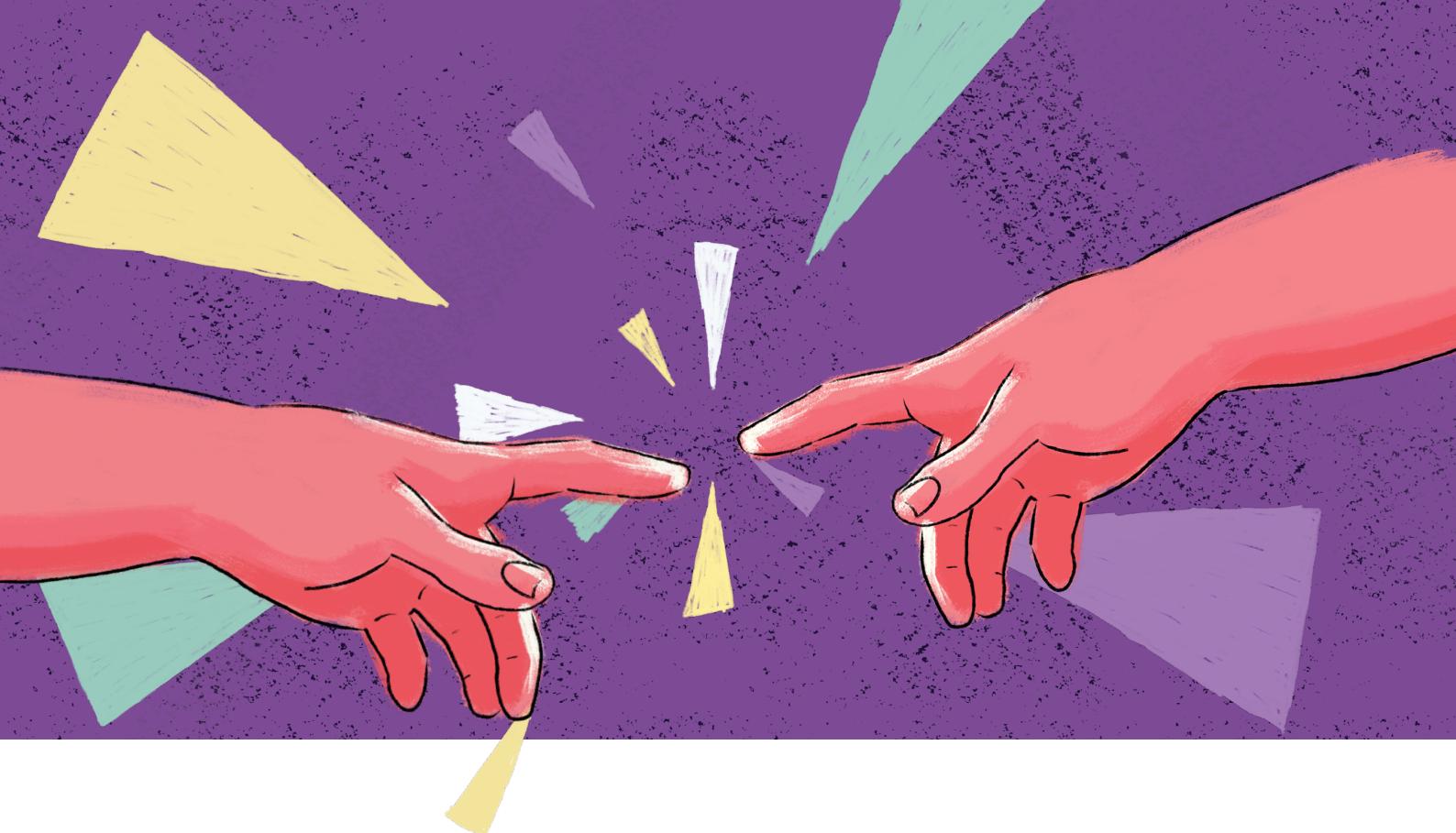
O que é Linguística Computacional?

A área de estudo de Linguística Computacional teve início no âmbito da Guerra Fria, em 1950, nos Estados Unidos. Naquele contexto, foram utilizados computadores para traduzir documentos que haviam sido escritos em outros idiomas, especificamente do russo para o inglês. Ainda naquela década, os estudos desse campo foram intensificados, embora seu maior crescimento tenha se dado a partir dos anos 1970.

Quem diria que essa iniciativa do passado ajudaria a moldar ferramentas como o ChatGPT, usado diariamente com propósitos diversos, da busca por bugs em um código até a tradução de textos de maneira instantânea?!

A Linguística Computacional tem como o objetivo principal fazer com que a linguagem natural possa ser processada computacionalmente. **Elá abarca os campos de Linguística, Informática e Inteligência Artificial**, sendo, portanto, uma área multidisciplinar, o que a torna propícia de ser estudada em um ambiente tão diverso como o Apple Developer Academy em termos de áreas do conhecimento com as quais temos contato ao longo da nossa formação.

Nesse aspecto, é importante mencionar o termo **Processamento de Linguagem Natural (PLN)**, que diz respeito à aplicação da Linguística Computacional junto a técnicas estatísticas de aprendizado de máquina. Existem muitos usos de PLN em nosso cotidiano, a citar, tradutores e sintetizadores de texto, agentes de conversação (**chatbots**), analisadores de sentimento e IAs Generativas.



Em PLN, destaco dois termos importantes: **tokenização** e **lematização**.

O processo de ***tokenização*** consistente, basicamente, em dividir um texto em pequenas partes menores, chamadas de tokens. Esses segmentos menores podem ser, por exemplo, cada um dos termos que compõem uma sentença, caso a **tokenização** seja feita por palavras. Adotando essa abordagem, a frase "O Swift é incrível" seria transformada nos tokens ["O", "Swift", "é", "incrível"].

Ao extrair **tokens**, é possível aplicar tratamentos, como remover pontuações e tornar tudo maiúsculo ou minúsculo. A adoção de tais práticas dependerá do objetivo em questão.

Suponhamos que você está usando PLN para analisar comentários recebidos no seu app publicado na App Store. Nesse caso, se um comentário possui muitos pontos de exclamação ("que app incrível!!!!"), pode haver um indicativo importante no contexto da análise dos sentimentos presentes naquela mensagem. Claramente a pessoa está muito empolgada com o seu aplicativo!

Ainda nesse âmbito, caso exista um comentário escrito completamente em letras maiúsculas ("ODIEI A FEATURE DE QUIZ"), a forma original é importante, pois revela claramente o sentimento de quem escreveu esse feedback (nesse caso, tem-se o sentimento de insatisfação com um aspecto do app).

Por outro lado, em mecanismos de busca ou filtros de texto, por exemplo, transformar para minúsculas simplifica o processo, já que "Sandbox" e "sandbox" serão tratados como o mesmo termo, evitando perda de resultados devido à capitalização.

Dessa forma, é importante analisar caso a caso quais modificadores devem ser aplicados ao realizar a **tokenização**.

A **lematização**, por outro lado, consiste em reduzir uma palavra à sua raiz. Isso significa, por exemplo, que os termos "chovendo", "chovido" e "choveu" serão transformados em "chover". Isso é útil, por exemplo, em um site de banco de imagens. Independentemente da forma digitada pelo usuário no campo de busca, o processo de **lematização** da entrada fará com que a imagem de chuva seja retornada como resultado.

PLN em Swift: desbravando o framework NaturalLanguage

No contexto de PLN, o framework **NaturalLanguage**, da Apple, fornece uma variedade de métodos úteis. Aplicando-o à linguagem de programação Swift, é possível segmentar (ou **tokenizar**, como vimos) um texto em parágrafos, frases ou palavras e obter informações sobre esses segmentos, como classe gramatical, idioma e sentimento presente na entrada fornecida.

O NaturalLanguage permite que a **tokenização** seja feita de 4 formas distintas: documento, parágrafo, sentença e palavra. Vamos, então, fazer essa segmentação usando o exemplo dado anteriormente. Para isso, precisamos:

- importar o **NaturalLanguage**;
- inicializar um **NLTokenizer** com a unidade **.word**, que indica que queremos dividir o texto em palavras;
- atribuir o texto de entrada ao tokenizador.

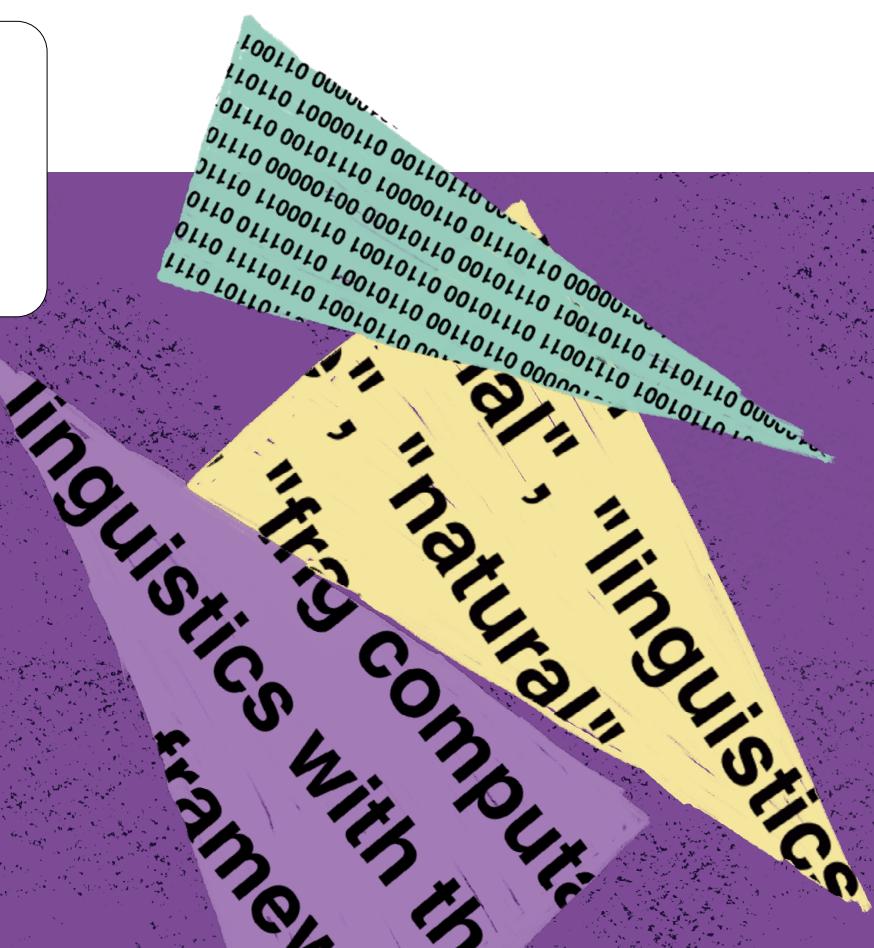
Por fim, para imprimirmos o resultado da

tokenização, usaremos o método `enumerateTokens(in:using:)` para iterar sobre cada token. Com base no intervalo (`tokenRange`) de cada token, extrairemos e exibiremos as palavras separadamente.

```
○ ○ ○  
tokenizer.enumerateTokens(in: text.startIndex..  
<text.endIndex) { tokenRange, _ in  
    let token = text[tokenRange]  
    print(token)  
    return true  
}
```

Como resultado, obteremos a nossa entrada segmentada, de modo que cada uma das palavras que compõem a sentença será, agora, uma parte menor, isto é, um token.

```
○ ○ ○  
import NaturalLanguage  
  
let text = "O Swift é incrível"  
let tokenizer = NLTokenizer(unit: .word)  
tokenizer.string = text
```



Agora, vamos para um exemplo um pouco mais avançado.

Imagine que você está construindo um app para ensinar Morfologia para o seu irmão mais novo...

Morfologia é como chamamos o estudo da estrutura, da formação e da classificação das palavras. Usando o NaturalLanguage, conseguimos determinar se um termo é um substantivo, um verbo ou qualquer outro constituinte de uma das classes gramaticais existentes. Para isso, vamos:

1. Criar uma instância de `NLTagger`, especificando `lexicalClass` como o esquema de tag a ser usado para definir a classe de palavras de cada termo e definir a propriedade `string` do analisador linguístico para o texto em linguagem natural de entrada. Nesse exemplo, vamos analisar a frase "I am happy".

○ ○ ○

```
import NaturalLanguage

let text = "I am happy"
let tagger = NLTagger(tagSchemes:
    [.lexicalClass])

tagger.string = text
```

2. Criar as opções para omitir pontuação e espaço em branco ao realizar a análise e enumerar toda a extensão da string, especificando `word` como a unidade de marcação e `lexicalClass` como o esquema de marcação. Além disso, no bloco de enumeração, vamos pegar uma substring do texto original em `tokenRange` para obter cada palavra.

○ ○ ○

```
let options: NLTagger.Options =
    [.omitPunctuation, .omitWhitespace]

tagger.enumerateTags(in: text.startIndex..
    <text.endIndex, unit: .word, scheme:
        if let tag = tag {
            print("\(text[tokenRange]):"
                \(tag.rawValue)")
        }
        return true
    }
```

Feito isso, teremos, como resultado, cada palavra acompanhada da sua classe.

Pronto, agora o seu irmão mais novo vai conseguir aprender Morfologia com a sua ajuda (e a do NaturalLanguage :p).

○ ○ ○

I: Pronoun
am: Verb
happy: Adjective

Eu AMO SWIFT

AMO SWIFT

AMO

Eu

AMO

Mais aplicações

Além das possibilidades de uso vistas acima, o NaturalLanguage possibilita muitas outras aplicações a serem exploradas. Alguns exemplos são:

- Identificador de nomes de pessoas, lugares e organizações;
- Analisador do idioma predominante em um texto;
- Classificador de sentimentos entre positivo, negativo e neutro;
- Identificador de palavras semelhantes em um texto;
- Classificador de assunto e contexto de conversas.

Próximos passos

Esse artigo buscou fornecer, brevemente, uma visão geral acerca das muitas possibilidades de uso do framework NaturalLanguage. A documentação oficial da Apple, disponível nas referências, fornece mais explicações, casos de uso e exemplos de código para muitas delas.

Se você ficou interessado em desbravar a Linguística Computacional a partir do Swift, recomendo fortemente que acesse. Deixo, ainda, recomendações de leitura associadas a esse tema:

- *Linguística Computacional: teoria e prática* de Gabriel Othero e Sérgio Menuzzi;
- *Estruturas sintáticas* de Noam Chomsky;
- *Linguagem e mente* de Noam Chomsky.

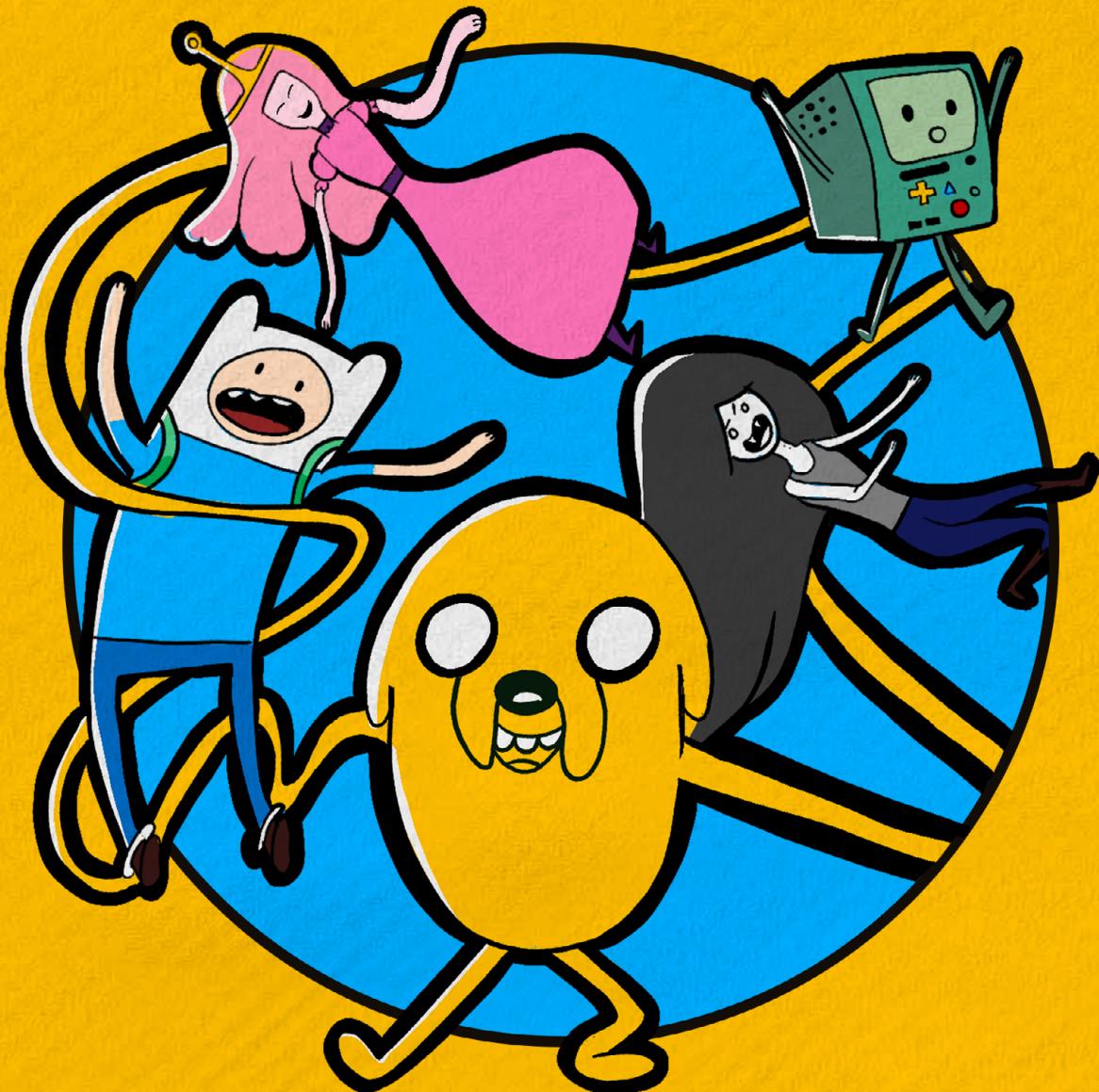
Há um universo de palavras e algoritmos a ser descoberto.

Te vejo em meio a essas letras (e códigos)! :)

Referências

APPLE DEVELOPER. NaturalLanguage. Disponível em: <https://developer.apple.com/documentation/naturallanguage/>. Acesso em: 31 out. 2024

OLIVEIRA, Anne Cleres da Silva; FRESCHI, Ana Carolina. LINGUÍSTICA COMPUTACIONAL: um mapeamento bibliográfico de 2000 a 2020. **Interface Tecnológica**. São Paulo, v. 17, n.2, p. 376-385, dez. 2020. Disponível em: <https://revista.fatectq.edu.br/interfacetechnologica/article/view/1056/548>. Acesso em: 31 jul. 2024.



DESIGN

TEXTO
MELISSA FREIRE

ILUSTRAÇÃO
LUCAS BAPTISTA

REVISÃO
MARCELLE RIBEIRO

ACESSIBILIDADE NA TERRA DE OOO: EXPLORANDO O DESIGN INCLUSIVO ATRAVÉS DE HORA DE AVENTURA.

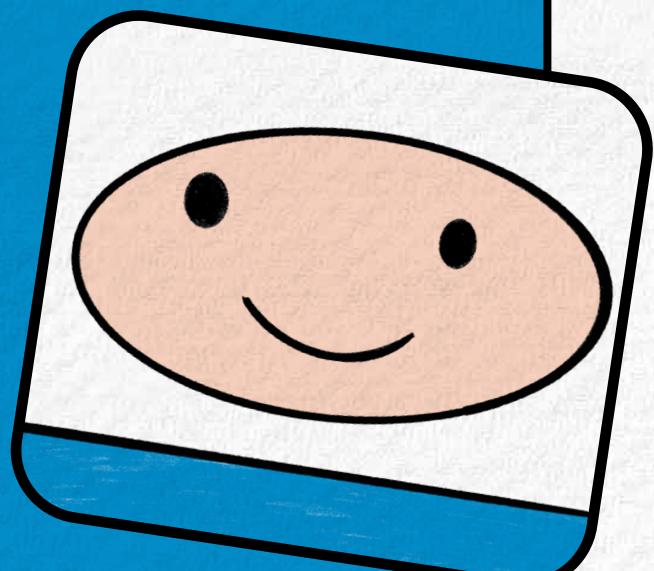
Se você assistia ao canal Cartoon Network ou SBT em meados de 2010, provavelmente você conhece o desenho "Hora de Aventura". Porém, se você ainda não teve a oportunidade de conhecer essa obra maravilhosa, aqui vai uma breve introdução:

Na encantada Terra de Ooo, continente onde se passa a história de Hora de Aventura, Finn, o último humano, e o seu fiel amigo, Jake, um cachorro com habilidades mágicas de mudança de forma, desbravam terras desconhecidas, em busca de novas aventuras épicas. Juntos, eles enfrentam desafios fantásticos sempre acompanhados de criatividade, humor e uma considerável dose de sensibilidade. Os personagens de Hora de Aventura, com as suas características únicas, conseguem ilustrar lições valiosas sobre questões do mundo real, inclusive no campo da acessibilidade no design de interfaces.

Da mesma forma como Finn e Jake desbravam o desconhecido, existem barreiras quando o assunto é criar aplicativos acessíveis e com uma interface intuitiva. Buscando quebrar essas barreiras, a Human Interface Guidelines (HIG) da Apple entra em ação, fornecendo diretrizes essenciais para a criação de interfaces acessíveis à grande maioria dos usuários. Ao juntar o universo épico de Hora de Aventura aos princípios de acessibilidade da HIG, é possível demonstrar de forma lúdica como as boas práticas de acessibilidade oferecidas pela HIG são essenciais para a criação de um design inclusivo.

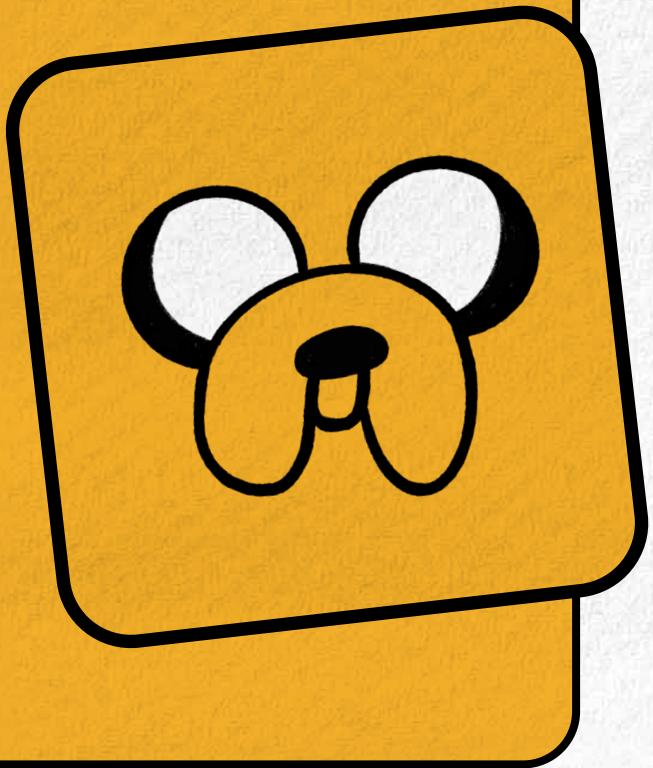
Simplicidade - Finn, O Humano

O Finn é objetivo e corajoso, e gosta de resolver desafios de forma prática e direta, e essas características do personagem podem ser associadas ao princípio da simplicidade no Design. Semelhante à maneira como Finn resolve as suas aventuras de forma simples e eficaz, um design simples torna as interações da aplicação fáceis e acessíveis. Se engana quem idealiza que uma interface bonita e eficiente é uma interface repleta de elementos chamativos e esteticamente bonitos, visto que a interface, com o intuito de ser acessível, precisa ser clara e objetiva, possibilitando que os usuários consigam navegar por ela de forma fluída, sem existir aquele sentimento de sobrecarga quando existe um fluxo difícil de ser compreendido pelo usuário, repleto de opções complexas a serem escolhidas.



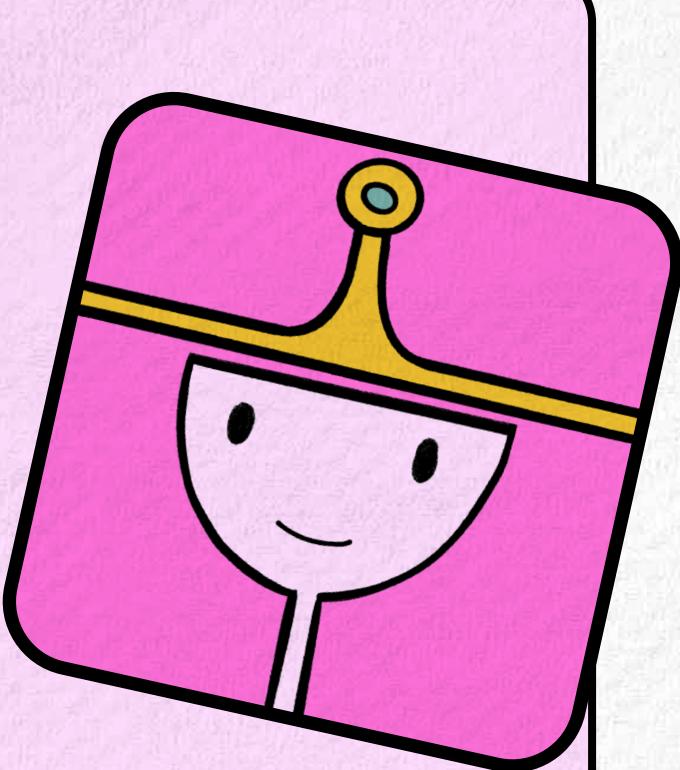
Perceptibilidade - Jake, O Cão

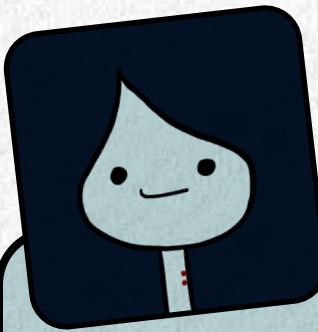
Jake tem uma habilidade mágica incrível de mudar de forma, podendo virar qualquer criatura que deseje, e essa característica notável dele reflete a questão da adaptação e da flexibilidade. De acordo com a HIG, a perceptibilidade consiste em fazer que todo o conteúdo da sua aplicação seja compreendido, independente se o usuário consome esse conteúdo de forma visual, de forma escutada ou através do toque. Dessa forma, assim como Jake se adapta a diversas formas diferentes, o conteúdo do seu aplicativo deve ser perceptível para usuários com diferentes necessidades, tais como aqueles que utilizam leitores de tela ou navegação através do toque.



Clareza e objetividade - Princesa Jujuba

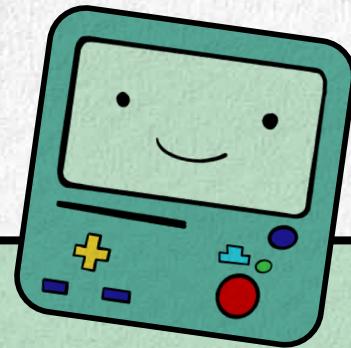
A Princesa Jujuba, conhecida por sua inteligência impressionante e pela sua paixão pela ciência, é um ótimo exemplo de uma pessoa clara e objetiva. Como governante da Terra de Ooo, ela busca realizar as suas ações de forma direta, sempre objetivando transmitir informações de forma clara, não abrindo espaços para dúvidas. Esses traços de personalidade da Princesa Jujuba esboçam muito bem o princípio de clareza e objetividade no design, o qual exprime que uma interface deve comunicar informações de uma maneira que seja compreensível para a grande maioria dos usuários, sem gerar uma sobrecarga de informação. Assim como a Princesa Jujuba garante uma comunicação clara com os cidadãos de seu reino, um design acessível também deve usar uma linguagem clara, instruções diretas e evitar a utilização de jargões, para promover uma experiência inclusiva e acessível.





Contraste e Visibilidade - Marceline, a Rainha dos Vampiros

Marceline, a Rainha dos Vampiros, possui um visual marcante, e chama atenção com o seu estilo único, sendo um ótimo exemplo para ilustrar o princípio de contraste e visibilidade no design acessível. Assim como Marceline se destaca na Terra de Ooo com as suas roupas escuras e a sua presença impactante, o contraste visual nas interfaces deve garantir que elementos importantes para a aplicação, tais como os botões e imagens, se destaquem uns dos outros, facilitando a leitura e a navegação da grande parcela dos usuários, especificamente aqueles com deficiências visuais. Portanto, o impacto visual de Marceline reflete a importância do contraste no design acessível, tornando a experiência mais clara e inclusiva para uma boa parte dos usuários.

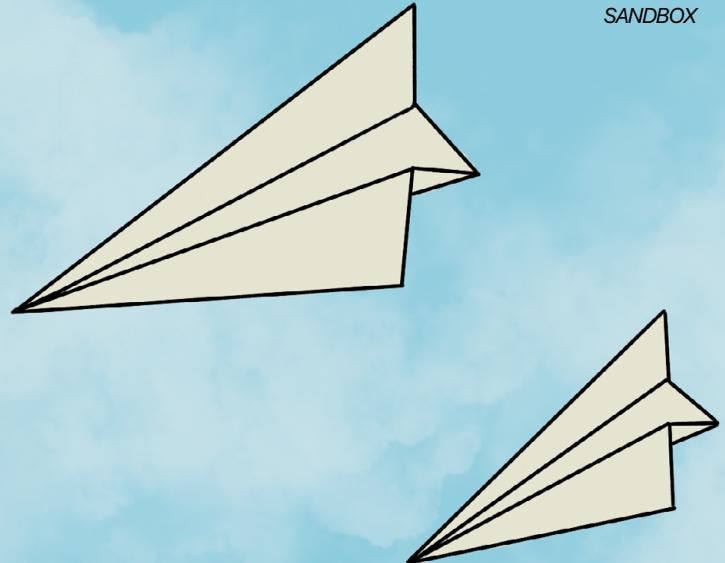


Recursos Assistivos - BMO

BMO é um pequeno videogame repleto de recursos e funcionalidades tecnológicas, sendo responsável por sempre ajudar Finn e Jake em suas aventuras mirabolantes. Da mesma forma que BMO auxilia Finn e Jake em seus desafios, as ferramentas de acessibilidade auxiliam inúmeros usuários no dia-a-dia, e podemos destacar o VoiceOver, recurso que fornece descrições audíveis do que está na tela, da Apple. Assim como BMO se adapta para ajudar os seus amigos em diferentes aventuras, é importante que o seu aplicativo também seja adaptável para ter acesso a esses recursos, buscando gerar um funcionamento inclusivo, garantindo, desse modo, uma ótima experiência para a maioria dos usuários da sua aplicação.

Ao combinar os princípios de acessibilidade da Human Interface Guidelines (HIG) com os personagens de Hora de Aventura, é possível entender, de forma lúdica e visual, a importância de criar um design inclusivo e acessível. O Finn ensina a valorizar a simplicidade e eficiência, Jake mostra a importância da adaptabilidade e a necessidade da perceptibilidade, enquanto a Princesa Jujuba representa a clareza e objetividade na comunicação, Marceline, com seu estilo marcante, destaca a importância do contraste e visibilidade, e BMO, com sua versatilidade, reflete a utilidade dos recursos assistivos.

Esses princípios, aplicados corretamente ao conceito de design de interfaces, são responsáveis por fazer que as aplicações digitais sejam mais inclusivas, garantindo que uma grande parte dos usuários, independentemente de suas limitações, possam ter uma experiência satisfatória e sem barreiras. Assim como em Hora de Aventura, onde cada personagem contribui com as suas características únicas para enfrentar desafios, um design acessível transforma a jornada digital dos usuários em uma experiência épica e sem obstáculos.



CÓDIGO

ENTENDENDO SWIFT CONCURRENCY

TEXTO

VICTOR MARTINS
(PUC-RIO)

ILUSTRAÇÃO

LAYS RIBEIRO

REVISÃO

KELLY LETÍCIA

Na WWDC de 2014, enquanto Craig Federighi (SVP of Software Engineering) apresentava Swift ao mundo pela primeira vez, ele destacou quatro características da linguagem: "Swift é rápida, moderna, segura e interativa". A Swift 6 traz um novo pilar de segurança para a linguagem, e é sobre essa característica que vamos sobrevoar.

Segurança, sobre a ótica de linguagens de programação, significa – em poucas palavras – que a linguagem tenta garantir que o código escrito nela resulte em execuções corretas, sem comportamentos inesperados, sem crashes e sem vulnerabilidades de segurança. Isso não quer dizer, por exemplo, que os aplicativos escritos em Swift não podem possuir vulnerabilidades de segurança. Significa que a linguagem se esforça para prevenir que isso aconteça.

Desde o seu lançamento, Swift conta com duas medidas de segurança muito marcantes: a segurança de tipagem (Type Safety) e a segurança de alocação de memória (Memory Safety).

Mesmo que talvez não tenha percebido, você já interagiu com ambas:

Quando tentou retornar uma String em um método cujo tipo de retorno era um Int e o compilador disparou um erro, estava lidando com a Type Safety. Ela previne que o seu programa entre em um estado inconsistente ao colocar um valor de um tipo num lugar onde se espera outro tipo.

Ao mesmo tempo, aposto que, mesmo que saiba o que são ponteiros, não se preocupa com eles ao programar em Swift. Isso acontece por causa das abstrações da linguagem, que fazem uso de ponteiros por baixo dos panos enquanto evitam que tenhamos acesso direto a eles. Isso nos exime de enfrentar os problemas comuns que os ponteiros causam em outras linguagens. Essas abstrações são parte dos esforços para alcançar a Memory Safety.



Mais Segurança?

Se os tipos das nossas variáveis já são verificados e o uso dos espaços de memória também, como podemos tornar Swift uma linguagem ainda mais segura?

É aí que entra o conceito de programação concorrente, que – em poucas palavras – é a capacidade de um programa de executar mais de uma tarefa ao mesmo tempo.

Os aplicativos que desenvolvemos fazem mais de uma coisa ao mesmo tempo em diversos momentos. Às vezes, mostramos uma barra de carregamento enquanto o conteúdo está sendo baixado da internet, mostramos uma animação divertida enquanto salvamos novas informações no servidor ou executamos um modelo do CoreML enquanto o usuário desenha. Na maioria dos casos, essas tarefas concorrentes são implementadas direta ou indiretamente por meio do Grand Central Dispatch, ou GCD.



De Onde Viemos

O Grand Central Dispatch, ou GCD, é um conjunto de tecnologias presentes em plataformas Apple que formam uma camada de abstração para facilitar a programação concorrente. Ele foi anunciado em 2009, e até hoje é amplamente utilizado por meio do framework Dispatch quando queremos executar mais de uma tarefa ao mesmo tempo nos nossos aplicativos.

É importante destacar que o GCD tem regras, restrições e garantias. Mas, sem um suporte oficial da linguagem para programação concorrente, tudo isso só pode ser verificado e garantido em tempo de execução.

Tempo de execução é o momento que o aplicativo está sendo executado. Ele é posterior ao tempo de compilação, que ocorre enquanto estamos programando.

Agora imagine a seguinte situação: estamos desenvolvendo um aplicativo antes dos recursos de Swift Concurrency serem adicionados à linguagem. Uma de suas telas mostra uma lista de receitas. Para carregar essa lista, cada receita é baixada individualmente de um servidor e, em seguida, adicionada em um Array, que é usado para popular a lista na interface exibida aos usuários.

Se cada receita demorar 1 segundo para carregar e carregarmos uma após a outra, sequencialmente, no caso de uma tela que contém 10 receitas, serão necessários 10 segundos de carregamento!

Suponha que, para acelerar esse processo, decidimos usar o GCD. Com ele, disparamos simultaneamente os 10 procedimentos de baixar as receitas e adicioná-las no Array. Consequentemente, preenchemos o Array de receitas em muito menos tempo, o nosso app está muito mais rápido e o feedback dos nossos usuários é maioritariamente positivo!

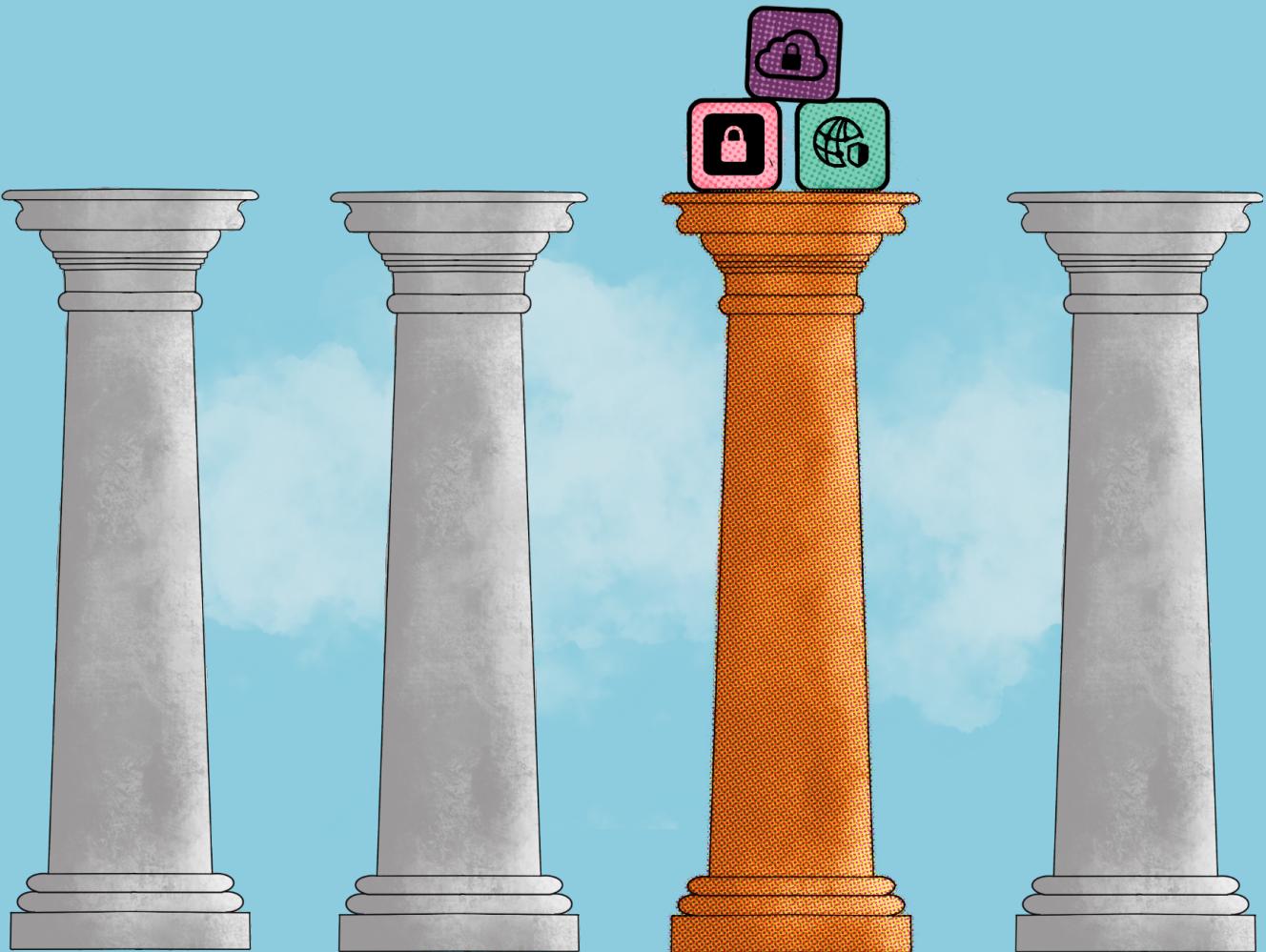


No entanto, alguns usuários estão reclamando que 1 ou 2 receitas às vezes não aparecem na lista. Por que isso está acontecendo?

Nesse momento nos deparamos com uma das dificuldades de lidar com a concorrência. As execuções concorrentes não são perfeitamente alinhadas toda vez. Pequenas variações sempre se fazem presentes. Mesmo em procedimentos disparados “ao mesmo tempo”, pode haver uma variação de alguns milissegundos no tempo em que cada um leva para começar. Além disso, no nosso exemplo, dependemos também da velocidade que o servidor responde cada uma das receitas, causando variações mais drásticas ainda.

Portanto, será que alguma variação nas execuções está causando esse problema relatado pelos usuários? Como investigar qual sequência de acontecimentos estaria causando problemas se não temos controle sobre essas variações? Para isso podemos tentar utilizar o Thread Sanitizer, ou TSan. Essa é uma das ferramentas de diagnóstico disponíveis no Xcode e ela serve exatamente para buscar problemas em execuções de código concorrentes.

Com a ajuda do TSan, descobrimos que na nossa implementação é um problema conhecido como data race. Ele acontece quando dois pedaços de código tentam alterar uma variável simultaneamente. No exemplo, ele pode acontecer quando duas receitas terminam de carregar ao mesmo tempo, já que isso faz com que elas sejam adicionadas ao Array de receitas ao mesmo tempo, o que não pode acontecer! Ao tentar fazer isso, uma das adições acaba sendo sobreescrita pela outra.



Mas, adicionar uma receita em um Array é algo que demora menos de 1 milissegundo, portanto, as chances de percebermos isso durante o desenvolvimento são muito baixas. Por isso que, mesmo com o nosso aplicativo sendo super popular, os usuários afetados são poucos e, quando o problema se manifesta, das 10 receitas, apenas 1 ou 2 não aparecem na lista final.

Notem como foi fácil criar um código que causa esse problema, e como foi difícil de perceber e resolver. Data races são um problema muito comum na programação concorrente. Essa investigação teria sido ainda mais difícil se tivéssemos um aplicativo mais complexo, com mais código concorrente e mais variáveis sendo alteradas.

Seria interessante se a linguagem fosse capaz de nos alertar quando o nosso código está propício a ter esse tipo de problema...

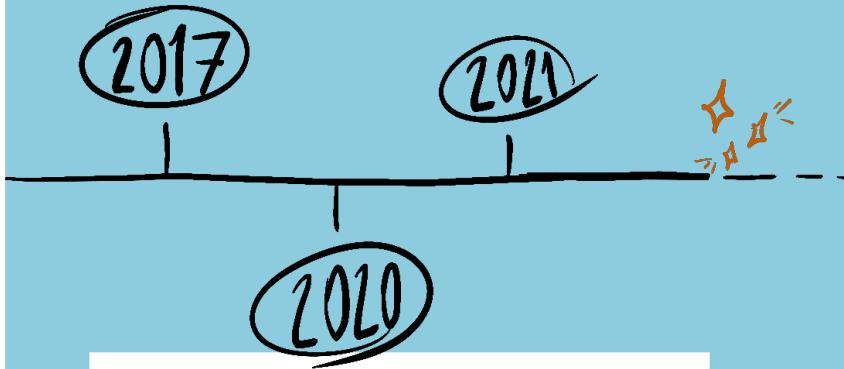
O Início de um Sonho

Como a segurança é um dos pilares de Swift, em 2017 o Chris Lattner (um dos engenheiros que participaram da criação de Swift) escreveu o Swift Concurrency Manifesto. Neste documento, ele descreveu as motivações pelas quais ele acreditava que a linguagem deveria ter recursos para expressar conceitos de programação concorrente. Esses recursos viriam acompanhados de verificações que, por exemplo, impediriam data races e isso traria uma série de benefícios para os usuários da linguagem. Ele pontuou também que colocar em prática um plano dessa magnitude seria um grande desafio. Até hoje, pouquíssimas linguagens com as características de Swift possuem os recursos propostos pelo manifesto.

Ao longo do tempo as discussões amadureceram e, em outubro de 2020, foi anunciado oficialmente o Swift Concurrency Roadmap: um roteiro contendo os passos que seriam tomados para levar recursos de programação concorrente para a linguagem e torná-la data-race safe. Acompanhando esse roteiro, estava um conjunto de Swift Evolution Proposals, onde cada uma delas detalhava um dos recursos a serem implementados, justificava a sua adição e fazia considerações sobre o impacto que ela causaria.

Esse anúncio, junto das propostas, deu início a longas e complexas discussões com centenas de postagens e participação ativa dos incontáveis membros da comunidade e dos membros do Core Team de Swift, que era a equipe responsável por esse processo.

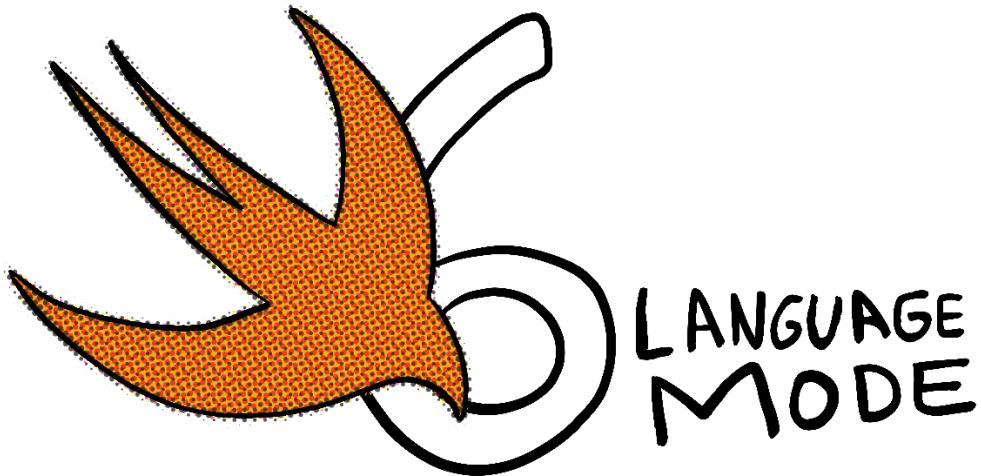
Finalmente, no Platforms State of the Union da WWDC de 2021, a Holly Borla (engenheira do compilador de Swift) anunciou que o Swift Concurrency estaria disponível no Swift 5.5 e a comunidade já poderia começar a se adaptar aos novos recursos.



O principal recurso de sintaxe adicionado na linguagem foi o par de keywords `async`/`await`. Para levar a concorrência não-estruturada para a linguagem – que é aquela em que não há uma relação de dependência entre a tarefa que foi disparada e o contexto que as disparou –, a API de Tasks foi adicionada. Por outro lado, dois conceitos, muito pouco explorados nas linguagens populares hoje, foram colocados em foco: os atores, explorados pela primeira vez na computação há décadas; e a concorrência estruturada, por meio da sintaxe `async let` utilizada em declarações.

Havia, no entanto, um ponto importante a ser destacado: a maioria das verificações que tornariam a linguagem data-race safe não estavam disponíveis. E no caso das verificações disponíveis, o Core Team optou por limitar os alertas ao nível de warning, e não erros, mesmo em casos graves.

Isso foi uma medida necessária porque nem todas as verificações estavam implementadas, mas principalmente por causa dos frequentes falsos positivos. Em muitos casos, a nossa intenção ao escrever um código concorrente era correta, mas não era possível expressar isso usando os novos recursos de uma maneira robusta o suficiente para satisfazer a linguagem. Por conta disso, supostos problemas eram apontados em códigos que na verdade não tinham problema algum. Para evitar confusões, essas verificações foram (em sua maioria) desabilitadas por padrão e precisavam ser ativadas manualmente.



Onde Estamos

De lá para cá, a comunidade absorveu os novos conceitos que fazem parte de Swift Concurrency e os colocou para uso na prática. Isso resultou em mais feedbacks, que também já foram incorporados à linguagem. Essas melhorias e a continuação da implementação dos recursos previstos no roadmap podem ser vistos nas mais de 12 propostas oficiais que foram discutidas e implementadas desde o anúncio. Esse processo de ouvir a comunidade fez com que pudéssemos expressar melhor as nossas intenções no código e trouxe diversas facilidades no uso dos novos recursos.

Na WWDC de 2024, vimos o anúncio de Swift 6, trazendo melhorias e mais recursos que programação concorrente. No Swift 6 language mode, a linguagem habilita todas as verificações de corretude de código concorrente, elevando o pilar de segurança e finalmente trazendo a data-race safety para a linguagem. Nas palavras da própria Holly Borla: "Swift 6 marca o início da jornada para tornar a data-race safety dramaticamente mais fácil."

Os projetos já existentes provavelmente contêm data races que nem percebemos. Precisaremos fazer anotações no código, utilizando a nova sintaxe, para permitir que a linguagem compreenda as nossas intenções com relação a acessos e alterações concorrentes. Por isso, eles precisam ativar o Swift 6 language mode manualmente para tirar proveito das verificações completas.

O processo de migração desses projetos inviavelmente envolve lidar com frameworks e packages que ainda não são compatíveis com os novos recursos. A linguagem nos permite marcar os módulos incompatíveis, fazendo com que o código que interage com eles não seja checado. Por um lado, perdemos os benefícios. Por outro, podemos avançar na migração enquanto aguardamos esses módulos se tornarem compatíveis.

A maioria dos frameworks da Apple já estão atualizados, como o HealthKit que foi usado como exemplo já em 2021. Frameworks mais complexos, como o SwiftUI, já são compatíveis, mas continuam recebendo sutis mudanças nas anotações até hoje.

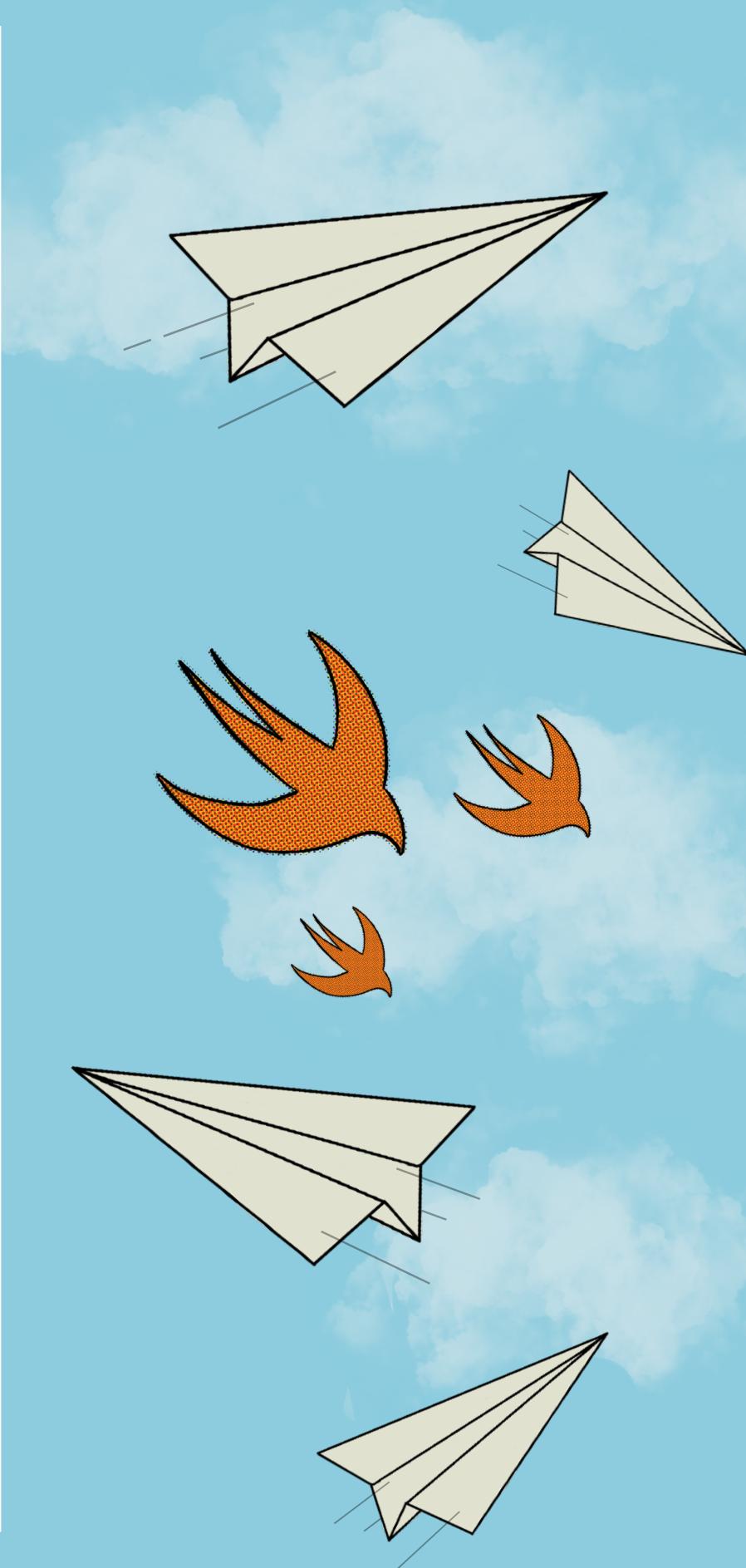
Para auxiliar na migração, a Apple fez uma session na WWDC 24 e publicou um guia com explicações sobre os novos conceitos, exemplos de erros mais comuns e como consertá-los, e uma sugestão de estratégia de migração incremental.

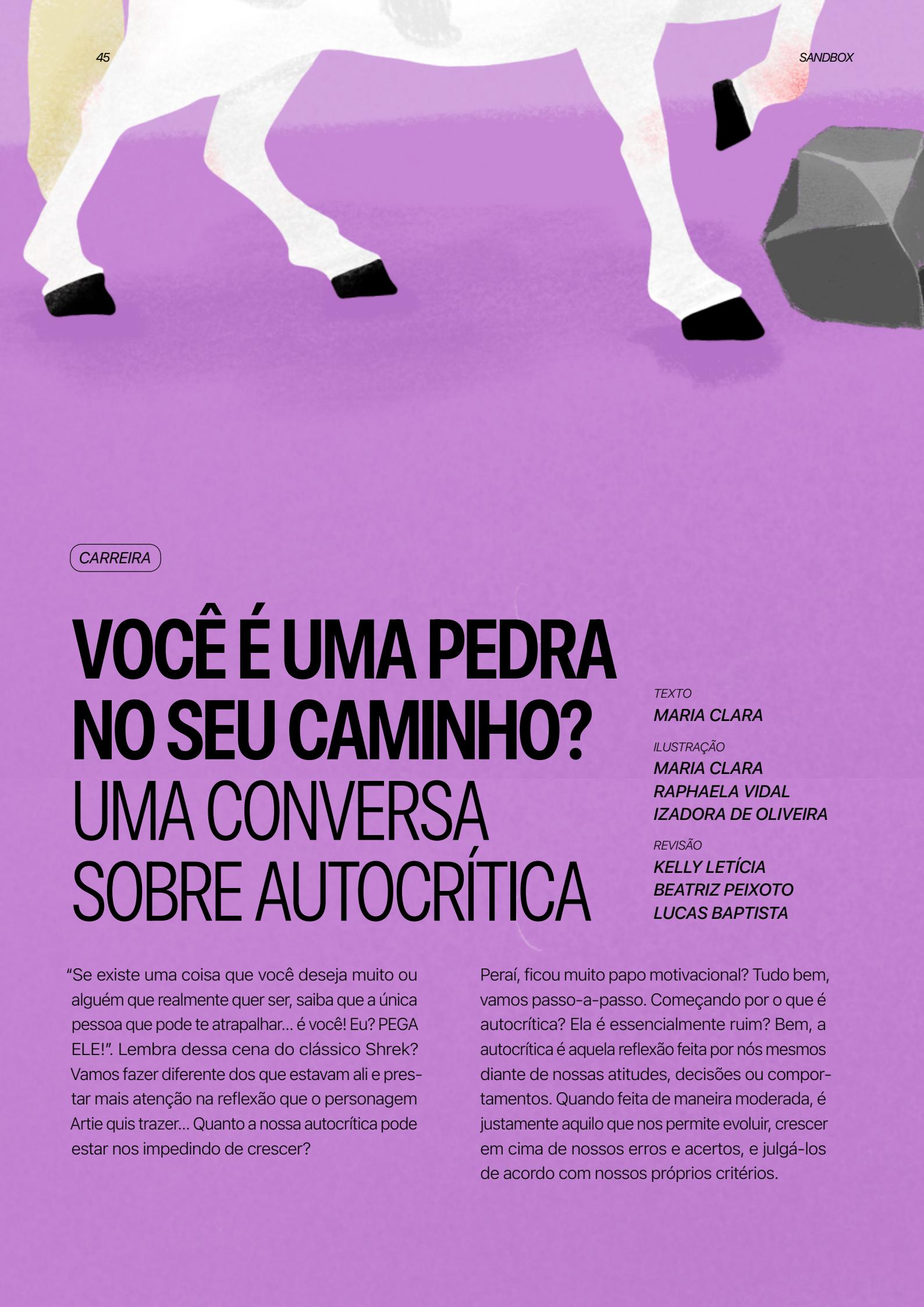
Para Onde Vamos

Com o passar do tempo, tanto os frameworks e Swift Packages que utilizamos, quanto os nossos aplicativos vão completar essa migração e poderão ser compilados no Swift 6 language mode. Isso os tornará mais seguros, agora contra data races, além de garantir o seu funcionamento correto da perspectiva da programação concorrente.

Dessa maneira, teremos a tranquilidade de saber que, ao utilizar recursos da linguagem para otimizar, por exemplo, o processo de carregamento da nossa lista de receitas, tudo funcionará corretamente. Isso porque se a nossa implementação tiver problemas que resultariam em data races, a própria linguagem nos avisará durante o desenvolvimento.

Essas garantias de segurança, além de serem positivas por si só, abrem portas para que mais desenvolvedores explorem recursos de programação concorrente que podem tornar os seus apps mais rápidos, responsivos e fluidos.





CARREIRA

VOCÊ É UMA PEDRA NO SEU CAMINHO? UMA CONVERSA SOBRE AUTOCRÍTICA

"Se existe uma coisa que você deseja muito ou alguém que realmente quer ser, saiba que a única pessoa que pode te atrapalhar... é você! Eu? PEGA ELE!". Lembra dessa cena do clássico Shrek? Vamos fazer diferente dos que estavam ali e prestar mais atenção na reflexão que o personagem Artie quis trazer... Quanto a nossa autocrítica pode estar nos impedindo de crescer?

Peraí, ficou muito papo motivacional? Tudo bem, vamos passo-a-passo. Começando por o que é autocrítica? Ela é essencialmente ruim? Bem, a autocrítica é aquela reflexão feita por nós mesmos diante de nossas atitudes, decisões ou comportamentos. Quando feita de maneira moderada, é justamente aquilo que nos permite evoluir, crescer em cima de nossos erros e acertos, e julgá-los de acordo com nossos próprios critérios.

TEXTO

MARIA CLARA

ILUSTRAÇÃO

MARIA CLARA

RAPHAELA VIDAL

IZADORA DE OLIVEIRA

REVISÃO

KELLY LETÍCIA

BEATRIZ PEIXOTO

LUCAS BAPTISTA

Porém, pode existir um momento na sua vida que essa voz, a qual deveria te ajudar a crescer, começa a te diminuir. O eco de um comentário maldoso, a comparação com o desempenho de outras pessoas ou o medo de julgamento... Todas essas questões externas começam a entrar na nossa cabeça e nos podar de dentro pra fora, até acreditarmos que o problema tá refletido no espelho.

Mas, há uma luz no fim do túnel! É possível diminuir o volume dessa vozinha chata na sua cabeça. Esse é um processo delicado, muitas vezes lento e, em casos mais extremos, deve ser acompanhado por profissionais. Mas, existem práticas que qualquer um pode começar ainda hoje para dar um chega pra lá nessa crítica desenfreada:

CONECTE-SE COM SUA HISTÓRIA

Pare um momento e pense. Como você chegou até aqui? Quem estava do seu lado? Que conquistas você teve? Que desafios você enfrentou? Tirar um tempo para pensar na sua história é um gesto poderoso que te ajuda a entender melhor quem você é e por que chegou onde está, uma atividade para aumentar a autocompreensão e identificar suas capacidades e suas dificuldades. Assim, ao invés do desempenho dos outros ou uma perfeição que não existe, você poderá utilizar esses fatos concretos para guiar sua autocritica de uma forma mais saudável.

Aproveite esse momento da melhor forma para você. Visite suas memórias, converse com pessoas queridas ou até construa uma linha da vida registrando tudo que você passou. Conecte-se com o aquilo que faz você não ser nem melhor nem pior, mas único: a sua história;

SAIA UM POUCO DA SUA CABEÇA

Como conversamos, nossa mente, com base em experiências passadas, pode acabar tendo uma visão distorcida ou enviesada do seu desempenho em algo. Mas você sabe né, o que pensa melhor que uma cabeça? Duas (ou mais)! Encontre pessoas de confiança, um mentor, um familiar ou um colega que possa te dar feedbacks construtivos do seu trabalho, te ajudando a sair dessa visão única a que você pode estar condicionado. Não se esqueça de analisar a aplicabilidade das críticas na sua visão, não leve tudo como verdade absoluta.



CULTIVE HOBBIES

Hobbies são tudo o que não são trabalho e que não devem sequer parecer um. Algo que você faz pelo puro prazer do processo, sem se importar com o resultado. Pode não ser fácil impedir que a autocobrança seja levada mesmo num contexto tão informal, mas é um bom momento para treinar, com paciência, essa habilidade de ser imperfeito, de se cobrar menos, sem a pressão de prazos ou de julgamentos alheios. Tudo isso enquanto se diverte e faz algo bonitinho! Busque se conhecer mais profundamente, experimente atividades que fujam da sua rotina e, caso não se sinta tão motivado, procure comunidades online ou presenciais para estimular esse hábito.



QUESTIONE MENOS, FAÇA MAIS!

Nesse último conselho, vou ser mais pessoal. Em mim, a coisa que mais engatilha a minha autocrítica desenfreada é pensar que estou no caminho errado. Que deveria estar estudando tal tecnologia, que já devia ter aprendido a fazer isso, muitas vezes baseado no que eu vejo dando certo nos outros. Porém, um dia, num momento muito áureo (one-o-one) uma alma muito sábia (a Gabi, ex-mentora aqui do Academy IFCE) me disse: "Não se questione se você está no lugar certo, mas faça o possível

para tornar o local onde você está, ser o certo para você" e isso tem me guiado fortemente desde então.

Não precisamos nos torturar se questionando se fazer tal coisa teria nos levado a um lugar melhor. Assim abrimos espaço para se orgulhar de fazer o que está ao nosso alcance para tornar o rumo que já estamos tomando ser o melhor possível, para nós mesmos, nos nossos critérios.

Papo difícil, né? A autocrítica não é algo fácil de lidar, ainda mais quando parece difícil encontrar a raiz desse julgamento interno. No entanto, é importante reconhecer que não é justo conosco mesmos nos impedir de arriscar e crescer em qualquer aspecto da nossa vida por causa dessa sensação de insegurança.

O processo de diminuição da autocrítica não é linear, nem do dia pra noite e não deve ser lugar para mais cobrança, seja paciente. Respire fundo e aplique essas pequenas práticas que conhecemos para ajudar a aliviar esse peso no peito e, quem sabe, assim como no **Shrek**, dar aquele grito interno de "PEGUE ELE!" para essa voz chata.



"A autocrítica só é útil quando acompanhada de autocompaixão."
— Kristin Neff

