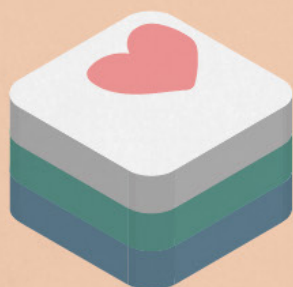
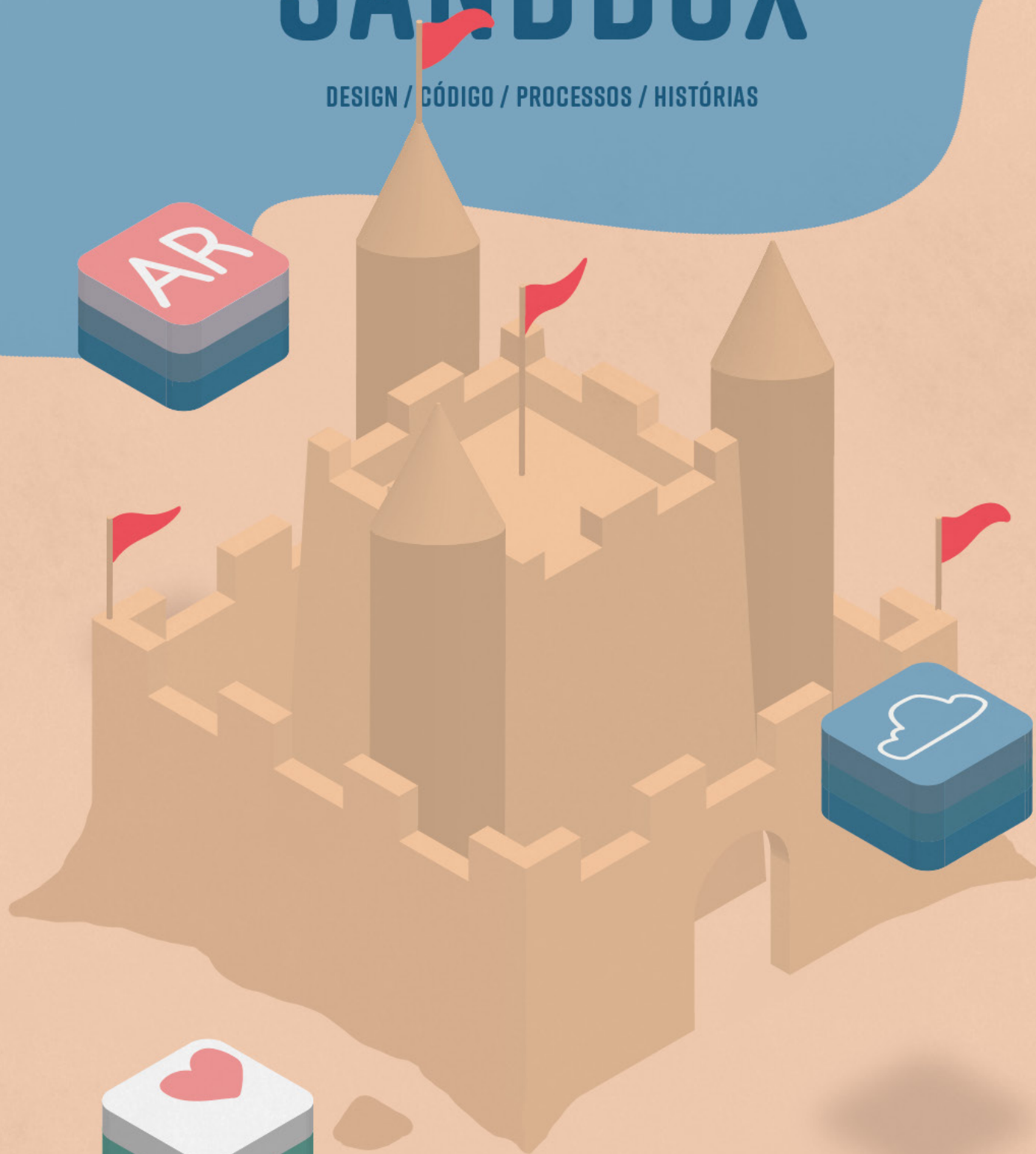


SANDBOX

DESIGN / CÓDIGO / PROCESSOS / HISTÓRIAS



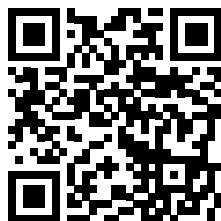
ACADEMY IFCE

Equipe de redação: Alley Pereira, Ana Karolina da Silva, Carolina Barbalho, Cecília Soares, Elias Ferreira, Fernando de Lucas Gomes, Gilberto Magno, Gustavo Lemos, José João Silva, Kellyane Nogueira, Mateus Nobre, Nathália Moura, Patrícia Sampaio, Ronaldo Gomes, Tales Conrado
Equipe de edição: Anderson de Alencar, Carolina Barbalho, Beatriz Carlos, Brenna Amorim, Cecília Soares, Fernando de Lucas Gomes, Helaine Pontes, Kellyane Nogueira, Nathália Moura, Tales Conrado, Vinícius Mesquita
Equipe de design: Carolina Barbalho, David Augusto, Fernando de Lucas Gomes
Equipe de Ilustração: Carolina Barbalho, David Augusto, Felipe Santana, José João Silva
Revisão técnica de código: Gabriela Bezerra, Mateus Rodrigues e Yuri Frota
Revisão técnica de design: Amanda Tavares e Carolina Barbalho
Projeto gráfico: Carolina Barbalho

Revista “Sandbox”. Edição 001. De Dezembro de 2020. “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 dos seus autores. **Para mais informações ou para solicitar licença sobre conteúdo,** entrar em contato com academy@lds.ifce.edu.br.

No momento da escrita das metérias desta revista, as versões mais recentes do **Xcode** e **Swift** eram **12 e 5.3** respectivamente. Consultar o histórico de atualizações caso esteja trabalho com uma versão diferente destas.

Para osaber mais sobre o
IFCE Apple Developer Academy, acesso o site:
www.developeracademy.ifce.edu.br



EDITORIAL

Tudo pois não tínhamos paredes suficiente...

Se, por algum motivo, você chegar agora no Academy IFCE, você vai encontrar laboratórios e mesas vazias, resquícios de um passado distante onde seres humanos iam para as universidades e trabalhavam sentados perto uns dos outros sem máscara - loucura total, eu sei.

Mas uma coisa não estará vazia: é muito provável que todas as colunas, principalmente do primeiro andar, estejam repletas de pôsteres, infográficos e *sheet cheats* feito por mentores ao longo dos anos. Mas aí que está, as cunhas estão tão cheias que um dia, quando fui tentar pregar um novo infográfico simplesmente não pude encontrar um único local livre para colocar o pôster.

Bem, talvez eu esteja exagerando um pouco, provavelmente existia algum espaço livre (na parede do banheiro feminino, por exemplo), mas de uma forma ou de outra, aquilo estava me incomodando. Deve haver uma forma melhor de organizar este tipo de material. E foi aí que veio o “E se”.

E se... a gente criasse um livreto para por todos estes materiais visuais juntos? Assim eles continuam sempre disponíveis nos laboratórios, mas nossas paredes ganham uma folga. Podemos até mesmo criar diversas edições de

livretos com os materiais criados ao longo dos dois anos e incluir materiais feitos pelos alunos. E já que vai ser um livreto mesmo, porque não incluir também matérias e outros conteúdo textuais mais densos como uma... revista. Não uma revista acadêmica, mas uma revista visual dessas de curiosidades, que o leitor tem prazer de folhear e ler.

Joguei a ideia para os outros mentores e eles se animaram, joguei para os alunos e eles compraram a ideia de cara. Então, comecei a pesquisar, montar um template de diagramação, pensar em como poderíamos organizar um projeto desses em conjunto com todos os alunos interessados.

Mas então veio a Pandemia (você achou mesmo que não iria ter essa frase em algum lugar?) e nossos esforços durante o primeiro semestre de 2020 se voltaram a adaptar a o Academy ao ensino remoto e garantir uma transição mais tranquila possível. Assim, o projeto voltou para a gaveta...

E ficou lá até agosto do mesmo ano, quando juntamos todos os alunos interessados novamente, dividimos papéis e finalmente abrimos inscrições de matérias.

Foi um grande sucesso! Vinte e uma submissões de ideias, das

quais dezoito foram pra frente e vieram parar nesta edição que está nas suas mãos agora.

E porque uma revista não é feita só de textos, vários alunos também se voluntariaram para ajudar com revisão, edição, design e ilustração!

E então começaram as reuniões do projeto. Nelas decidimos a voz da revista, planejamos workshops de ilustração e redação para todos os interessados, distribuimos tarefas, mandamos mensagens aperreando os amigos para eles terminarem seus textos e passamos incontáveis minutos conversando besteira antes de todos chegarem nas benditas reuniões de quartas-feira à tarde.

E por isso, para cada uma das pessoas envolvidas no projeto, tudo que posso dizer é: obrigada! Obrigada por embarcarem nessas ideias doidas e fazer delas um projeto de vocês também.

Cada um deu o seu melhor e é só por isso que essa revista maravilhosa existe hoje. Nem todos nós conseguimos terminar nossos textos, fazer todas as ilustrações que planejamos ou comparecer a todas as reuniões que gostaríamos, mas isso não faz nossos esforços menos importantes. Vocês são demais!

Por fim, boa leitura!

Carolina Barbalho
EDITORA GERAL

CONTEÚDO {



↳ NUNCA É TARDE PARA APRENDER O NOVO

Sua idade não deve ser um empecilho para seguir seus sonhos.

08 CORES E ACESSIBILIDADE

A interface é apenas a superfície de um aplicativo!

10 MACHINE LEARNING: UM OCEANO DE POSSIBILIDADES

Mergulhe no mundo de machine learning com esse tutorial



12 SORRIAM E ACENEM, RAPAZES

Um relato sobre timidez, confiança e autonehcimento.

14 ICEBERG DA UX

A interface é apenas a superfície de um aplicativo!

16 O QUE OS OLHOS VEEM, O CORAÇÃO SENTE

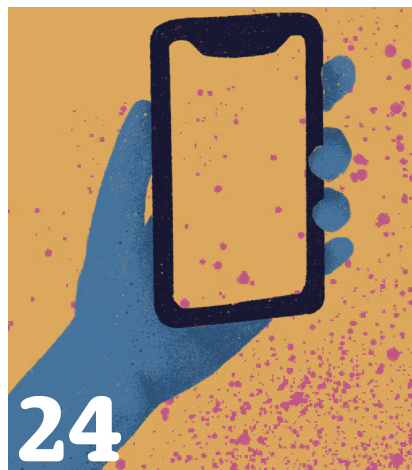
Qual o impacto do design na nossa vida e experiências?

20 POR QUE NÃO USAR CONSTRAINT LEFT AND RIGHT?

Muitos perguntaram, aqui está a resposta!

22 FLUXO DE DESENVOLVIMENTO COM GITFLOW

Um infográfico sobre como desenvolver com o Gitflow.



↳ GAMEIFICAÇÃO PARA APPS

Como agregar elementos de jogos em aplicativos

26 DESBRAVANDO MAC GAMES

Tudo que você precisa saber para iniciar seus jogos para macs.

28 NAVEGANDO COM COORDINATORS NO XCODE

Um guia rápido sobre o uso de coordinators no Xcode.

32 DE CODER A DEVINER

Conheça Amanda e Levy, Alumni do Academy IFCE.

36 TRABALHO E FAMILIA

Equilíbrio é tudo na vida... e quando entra o trabalho?



↳ MERGULHANDO NA WEB

Dicas e técnicas para você não se afogar mais nas pesquisas!

40 POR QUE EU DEVO ME PREOCUPAR COM A LGPD?

Segurança e privacidade são cada vez mais importantes.

42 MULHERES NA TECNOLOGIA

Conheça algumas das mais incríveis mulheres da tecnologia!



↳ QUAL A IMPORTÂNCIA DOS TESTES DE USUÁRIO?

As vezes colocado de lado, testes são fundamentais, vem ver!

48 2020 EM NÚMEROS

HISTÓRIAS {

Nunca é tarde para aprender o novo

TEXTO *Patrícia*

EDIÇÃO *Anderson*

DESIGN *Carolina*

ILUSTRAÇÃO *Carolina*

A inspiração para esse texto veio das diversas vezes as quais ouvi de pessoas jovens que seria muito tarde para estudar determinado assunto ou que seria muito tarde para ingressar em uma nova carreira. É imprescindível que os jovens também entrem nessa nossa conversa, pois aos que estão ansiosos para conquistar seu espaço profissional, também podem se valer da experiência dos que já deixaram seus projetos iniciais para mudar e traçar novos caminhos profissionais.

Não importa aqui qual o seu desejo de mudança, nunca será tarde demais para começar tudo novamente. Então, devemos encarar a vida como em constante recomeço e constatar que a derrota não existe para quem sempre quer seguir adiante. Portanto, as pedras que hoje estão em seu caminho podem formar sua calçada amanhã.

Na tecnologia temos exemplos fantásticos como Robert Noyce que criou a Intel com 41 anos, Jan Koum que lançou o WhatsApp ou com o Jimmy Wales com a Wikipedia que em ambos os casos tinham 35 anos quando tiveram sucesso em seus projetos. Percebemos que a idade biológica em nada vai interferir nos seus sonhos e acredite, só você mesmo pode limitar sua caminhada. As histórias bem-sucedidas têm algo em comum, existe um momento que você deve filtrar os “conselhos”. Vão existir todo tipo de críticas e conselhos nessa sua jornada e somente você poderá distinguir o efeito das

opiniões em sua vida.

Um outro exemplo é a fabulosa Masako Wakamiya. Ela obteve seu primeiro computador pessoal aos 58 anos, pouco antes de se aposentar de um banco. Mal sabia ela que estava começando uma jornada que a tornaria uma desenvolvedora de aplicativos para iPhone conhecida, com 81 anos. Hoje com 85 anos, Wakamiya se autodenomina uma evangelista de TI e incentiva outros idosos a usar a tecnologia digital para enriquecer suas vidas. Ela escreve livros enquanto espalha sua mensagem no circuito de palestras no Japão e no exterior.

NÃO IMPORTA QUAL O SEU DESEJO DE MUDANÇA, NUNCA SERÁ TARDE DE MAIS PARA COMEÇAR TUDO NOVAMENTE

Masako é a criadora do aplicativo Hinadan. O produto é um game baseado em um festival japonês chamado Hinamatsuri (dia das bonecas). O Hinamatsuri é celebrado normalmente dia 3 de março e bonecas tradicionais do país são dispostas em degraus. Falar de Masako é falar sobre inclusão digital em várias esferas, é mostrar que o tempo não é o fator limitante para nada e constatar que quando você faz o que gosta, não existem barreiras.

As vezes precisamos reescrever uma antiga

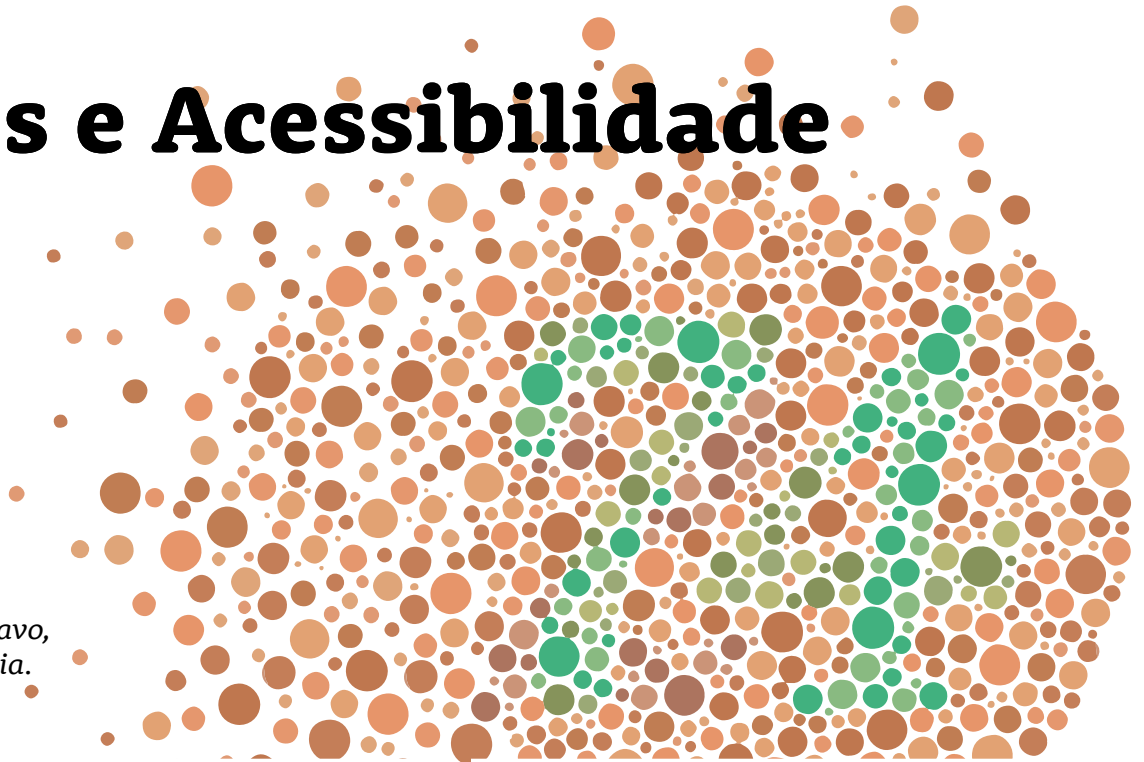


história e modificar o início, o meio, para ter seu belo final com disposição e paciência. Celebrar cada conquista como um passo a mais, um degrau a mais nessa escalada. Para tanto, precisamos muito desse fôlego para estar sempre seguindo em frente em busca de nossos sonhos.

Portanto, se você que está aqui lendo esse artigo e está começando a vida profissional agora, inspire-se nos exemplos bem-sucedidos para aumentar sua resiliência, sua paciência, sua fé e sua confiança. No entanto se quem está lendo se considera “velho demais”, peço somente que reflita sobre os exemplos acima e conclua que é possível recomeçar sempre. Nunca será do zero, mas em uma nova trajetória com sabedoria e autocohecimento. ❶



Cores e Acessibilidade



TEXTO *Elias, Gustavo, Nathália e Cecília.*
EDIÇÃO *Carolina*
DESIGN *Carolina*

O que você vê nesta imagem? Há diferentes respostas para essa pergunta, pois nem todo mundo enxerga da mesma maneira.

Algumas pessoas não são capazes de distinguir todos os elementos representados. Isso afeta principalmente pessoas com uma condição chamada Daltonismo. Uma deficiência visual que tem como principal característica a dificuldade no reconhecimento de cores, principalmente a distinção entre o vermelho e o verde, mas que pode se apresentar de outras formas.

Geralmente, esta doença é uma herança genética que acompanha o indivíduo desde o seu nascimento e é muito mais comum entre pessoas do sexo masculino. Estima-se que 1 em 12 homens são daltônicos e somente 1 em 200 mulheres são daltônicas.

O daltonismo possui diversos tipos, que são classificados de acordo com a localização e a maneira que a deficiência se manifesta. A forma com que cada pessoa daltônica vê, depende do seu tipo de daltonismo.

TIPOS DE DALTONISMO

Veja como as cores se manifestam em alguns dos tipos mais comuns de daltonismo



Sem daltonismo



Protanopia



Deuteranopia



Tritanopia

Design para daltônicos

Em uma tela, onde muitas interações são movidas por estímulos visuais dos usuários, é preciso atentar-se na maneira como esses estímulos podem ser interpretados por pessoas daltônicas.

Um designer de interfaces deve pensar em cada aspecto de uma interação, ou seja, tudo deve estar lá por uma razão e para causar uma série de sensações e estímulos planejados.

Um bom designer deve manter sempre em mente as dificuldades que algumas pessoas podem possuir ao acessar sua interface.

Ainda que muitas vezes a acessibilidade não seja uma pauta do projeto, sem este tipo de planejamento, diversos públicos podem ser prejudicados. Um desses públicos sendo as pessoas daltônicas.

Infelizmente, muitas empresas não levam o daltonismo em consideração por ser visto como algo de pequena relevância, devido ao público daltônico ser pequeno. E, por mais prejudicial que seja, essa negligência costuma ocorrer frequentemente. E para piorar, muitas vezes falta ao designer conhecimento e exposição ao assunto.

Mas as soluções para interfaces mais acessíveis a daltônicos

são na verdade muito simples. Existem diversas ferramentas para auxiliar os desenvolvedores, como por exemplo o *Palleton*, que auxilia a esimular a visão de um daltônico. O que permite ao designer testar em tempo real, se determinados elementos estão compreensíveis.

Além disso, podemos desenvolver um design inclusivo simplesmente observando as dificuldades que os usuário podem ter e buscando soluções para elas. A seguir listamos algumas boas práticas que, apesar de simples, são a base para transformar qualquer aplicativo em um produto acessível a daltônicos. ❶

1

ATENTE-SE AO CONTRASTE

Alguns contrastes baseados em cores podem não ser suficiente para pessoas daltônicas.



errado :(



agora sim!

2

NÃO USE APENAS CORES

Use cores para reforçar uma informação, mas nunca utilize como único indicador de algum estado.



errado :(

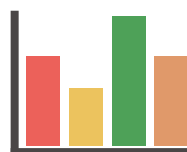


agora sim!

3

ADOpte TEXTURAS

Texturas são um estímulo visual que estará sempre visível, mesmo para os daltônicos



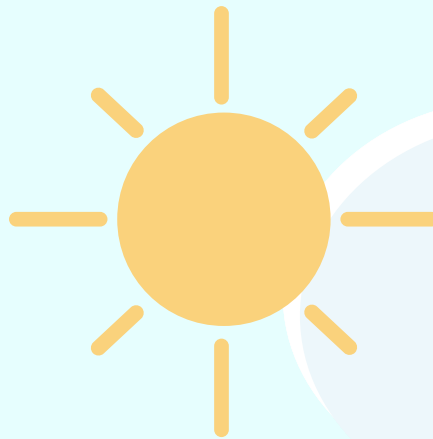
errado :(



agora sim!

CÓDIGO {

Machine Learning: Um Oceano de Oportunidades



TEXTO *Ronaldo*
EDIÇÃO *Fernando*
ILUSTRAÇÃO *Fernando*
DESIGN *Fernando*

Machine Learning é um subcampo da inteligência artificial em que, através do fornecimento de dados e aplicação de cálculos matemáticos e estatísticos, o computador encontra padrões entre os dados. Basicamente, a máquina é “treinada” usando uma quantidade grande de dados e algoritmos que dão à ela a habilidade de aprender sem serem explicitamente programados. O objetivo desse processo é gerar um modelo matemático que é usado para fazer previsões a partir de novos conjuntos de dados.

Nos últimos anos, o crescimento na produção de dados tem impulsionado a área de Machine Learning. Como resposta à isso,

a Apple vem desenvolvendo ferramentas para auxiliar desenvolvedores que desejam aplicar conceitos de Machine Learning em seus aplicativos. Uma dessas ferramentas é o Create ML, com o qual é possível criar modelos de Machine Learning com apenas alguns cliques e adicioná-lo ao código da sua aplicação em Swift.

Machine learning no mundo iOS

Como dito anteriormente, a Apple vem desenvolvendo ferramentas que auxiliam os desenvolvedores a trabalhar com Machine Le-

Create ML na Prática

A

O Create ML vem incluso quando você instala o Xcode. Aqui está o passo a passo para criar um modelo de classificação de imagens novo para usar na sua aplicação.

Já que iremos criar um modelo de classificação de imagens selecione “Image Classification”. Depois, nomeie o projeto fazendo uma breve descrição sobre o modelo e seus objetivos.

B

Após isso são adicionados os dados de treinamento. O Create ML determina automaticamente o número de classes baseado no número de subdiretórios da pasta com os dados de treinamento.

É importante que suas classes tenham um número próximo de dados, para que não haja discrepância.

arning. Na WWDC de 2019, foi anunciada uma série de novidades sobre Machine Learning, como o framework Core ML 3 que permite o uso de modelos pre-treinados, aproveitando o hardware dos dispositivos ao máximo. Nesse mesmo ano, também foi anunciado o aplicativo Create ML que permite a criação de modelos Core ML através de uma interface gráfica sem precisar escrever nenhuma linha de código. Além do Core ML, a Apple fornece outros frameworks que facilitam o uso de Machine Learning como o Vision para visão computacional, Natural Language para processamento de linguagem natural, dentre outros para processamento de voz e som. Isso permite que os desenvolvedores possam aplicar poderosos modelos em seus aplicativos sem precisar recorrer a outras linguagens de programação ou ferramentas de terceiros. Além disso, esses frameworks são intuitivos, fáceis de usar e, na maioria dos casos, os desenvolvedores não precisam saber ao fundo a teoria por trás dos modelos para aplicá-los no código. Mesmo com essa facilidade, ainda existe o Core

ML Tools, que permite a conversão de modelos dos mais famosos frameworks, como TensorFlow, PyTorch, Keras e scikit-learn para modelos CoreML que podem ser usados em aplicativos iOS.

Create ML

A geração de modelos de Machine Learning exige conhecimentos avançados sobre o tema. Como solução para isso, o Create ML foi desenvolvido para gerar rapidamente modelos de Machine Learning através de uma interface gráfica. O Create ML, dispensa conhecimentos avançados em Machine Learning permitindo com que rapidamente você gere e insira os modelos em sua aplicação Swift. Isso facilita a implementação de Machine Learning em aplicações iOS e ainda abre um potencial gigantesco para seu uso. Atualmente, o Create ML conta com ferramentas para treinamento de classificador de imagens, detecção de objetos, classificador de ações corporais, classificação de som, classificação de texto e de dados tabulares. Cada fer-

O CREATE ML NOS PERMITE A APLICAÇÃO DE PODEROSOS MODELOS DE ML COM MUITO MAIS FACILIDADE

ramenta oferece parâmetros personalizados, com algumas individualidades e customizações, porém o básico do processo geral é o mesmo.

Futuro

O uso de Machine Learning permite um mundo de oportunidades unindo a capacidade de aprendizado ao poder operacional de um computador. Com todas essas ferramentas e a facilidade do Create ML, não tem porquê não investir mais tempo e estudo de Machine Learning em aplicações iOS. O Create ML veio para facilitar a aplicação de Machine Learning em dispositivos iOS, isso torna o mundo do Machine Learning mais próximo do desenvolvedor e permite que ele mergulhe nesse oceano de oportunidades. ⓘ

C

Então são ajustados os parâmetros, podendo utilizar a configuração padrão ou adicionar interações e parâmetros para otimizar o modelo. Após isso, selecionamos a opção “Train” para treinar o modelo.

Dependendo dos seus parâmetros o tempo para o treinamento pode variar um pouco.

D

Logo em seguida é disponibilizada a acurácia do modelo. Você pode testar a acurácia do modelo na aba “Preview”. Para isso forneça dados diferentes dos dados de treinamento.

No programa, na aba Evaluation é possível observar a acurácia de treino, validação e teste.

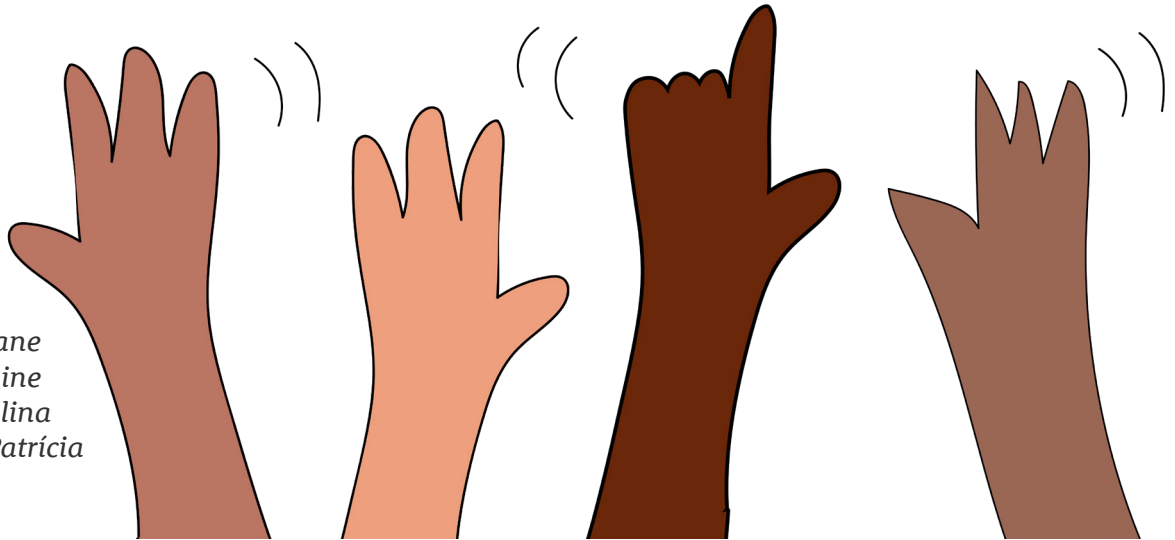
E

Por fim, modelo Core ML estará na aba “Output”, com todas as suas informações. Agora é só arrastá-lo para o projeto no Xcode e usá-lo.

HISTÓRIAS {

Sorriam e acenem, rapazes!

TEXTO Kellyane
EDIÇÃO Helaine
DESIGN Carolina
ILUSTRAÇÃO Patrícia



Sempre admirei aquelas pessoas despoçadas e corajosas, que enfrentam o público para mostrar suas ideias, fazem coisas engraçadas sem medo de serem julgadas e que inspiram outras pessoas a serem elas mesmas. De repente, me vejo pensando se um dia eu poderia ser assim...

Qual o segredo para deixar a timidez de lado? Será mesmo que posso ser boa o bastante no que eu faço? Qual a sensação de não ter medo? Há problema em ser tímido?

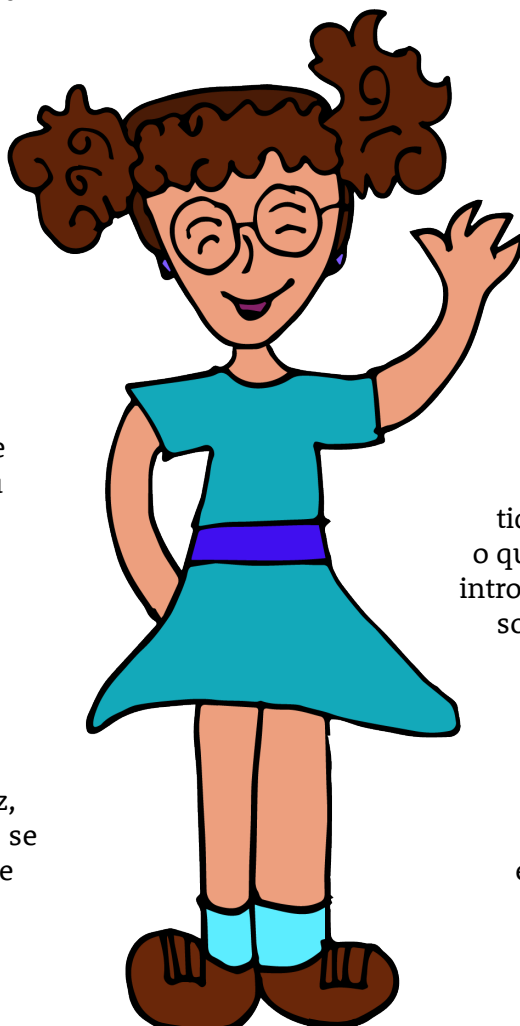
Durante alguns meses estive pesquisando sobre o assunto, observando comportamentos e buscando entender se as pessoas nascem tímidas ou se o mundo as transforma. Então, resolvi compartilhar com vocês o que descobri durante minha busca por respostas.

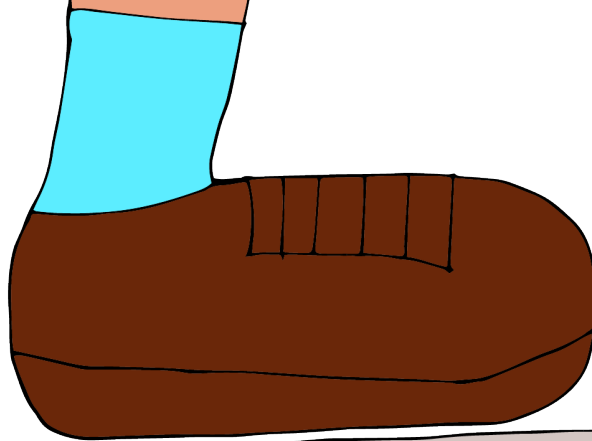
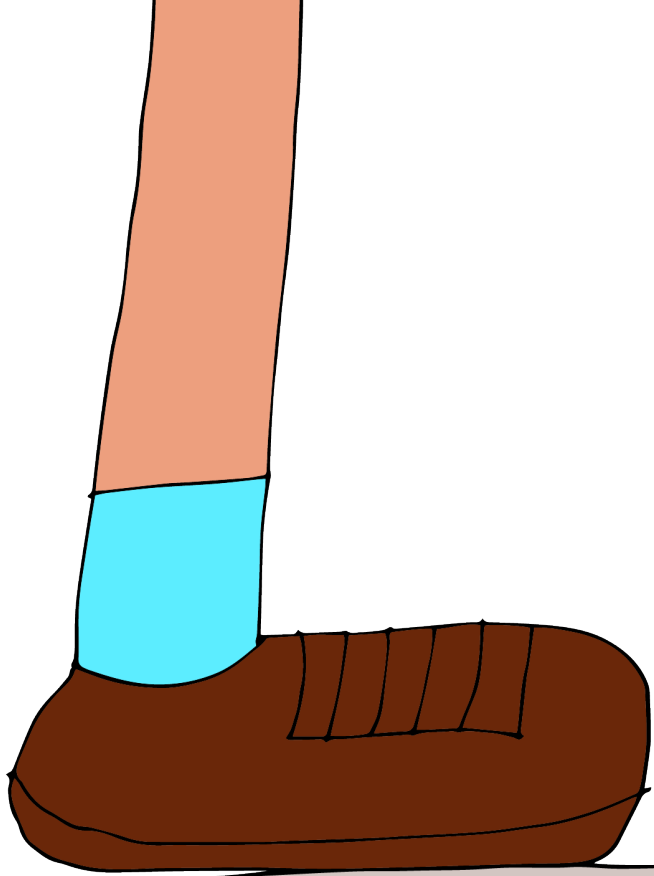
Primeiro, fui em busca do autoconhecimento. Não é possível colocar todo mundo em um pacote e só generalizar, não é mesmo?! Com a timidez, também há muitas coisas a se considerar. Há psicólogos que acreditam que todo mundo

nasce um pouco tímido e aos poucos aprende a lidar com isso. A timidez também pode ser um traço de personalidade que não é de todo ruim. Logo, a timidez em si não é um problema, desde que não te impeça de realizar coisas importantes e/ou prejudique seus relacionamentos.

Entender a timidez também pode ser bem confuso. Por vezes tratei a introversão como timidez e vice-versa. De fato, cometi um equívoco. Embora constantemente encontre timidez e introversão como sinônimos, há diferença entre elas.

Explicando melhor, alguns tímidos também são introvertidos mas isso não é uma regra. E o que é introversão afinal? Bom, a introversão é caracterizada por pessoas que estão mais atentas ao seu mundo interior, diferente da extroversão em que as pessoas tendem a se orientar pelo mundo ao redor delas. Os introvertidos geralmente gastam mais energia quando estão interagindo e por isso sentem mais a necessidade





Por mais que a gente queira, o processo de autonehciamento não é fácil, muito menos rápido. É uma jornada que dura a vida toda!

de terem seus próprios momentos de tranquilidade para “recarregá-las”, enquanto os extrovertidos ganham energia com a interação. Uma forma errada de ver a introversão é dizer que os introvertidos não gostam de estar com as pessoas, não é nada disso! Apenas, eles tendem a ter mais momentos de reflexão do que de interação. Dito isto, é normal encontrar tímidos extrovertidos. Entender isso trouxe muitas das respostas de que eu precisava.

Seja como for, nenhum tímido está isento dos friozinhos na barriga, dos tremeliques nas mãos ou aquela bela gaguejada em público. Ser o centro das atenções é algo muito indesejado pois normalmente quem é tímido tem medo de ser julgado ou avaliado e tende a pensar coisas negativas sobre si mesmo. Como por exemplo, não ser capaz de realizar alguma tarefa ou se sentir travado em ocasiões importantes. Então,

A timidez faz parte de quem você é, e só passa a ser um problema se estiver te impedindo de realizar coisas importantes na sua vida.



quando a timidez começa a ser prejudicial podemos aprender hábitos saudáveis para lidar com ela. “Os tímidos de sucesso não mudam quem são. Mudam seu modo de pensar e suas ações.”

Uma dica é encarar a timidez de uma ótica positiva. Uma coisa muito legal sobre os tímidos é que eles podem ser ótimos líderes devido a sua natureza cuidadosa e reflexiva, geralmente são bons ouvintes e são empáticos porque conseguem entender a dificuldade do outro em compartilhar publicamente alguma ideia. Também pode ajudar bastante lembrar que todos somos imperfeitos e que está tudo bem se as coisas não saírem como imaginamos. Tenha certeza de que você pode vencer as limitações, seja um seminário com inúmeras pessoas ou um simples encontro com alguém especial, “sorria e acene” e nunca deixe de se desafiar mesmo quando sentir medo. ❶

INFOGRÁFICO {

O Iceberg da UX

TEXTO *Carolina*
EDIÇÃO *Kellyane*
DESIGN *Carolina*
ILUSTRAÇÃO *Carolina*



Todo produto deve entregar algo de valor ao usuário.



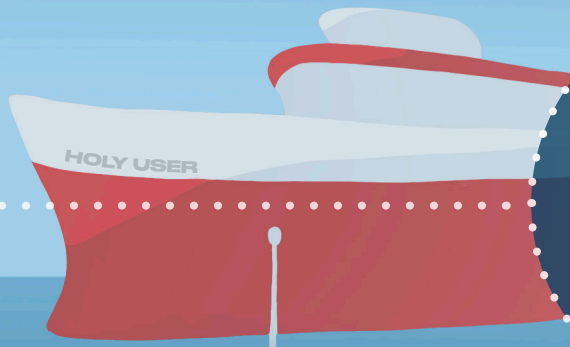
Design não é só tela bonita. Se você já teve tempo para estudar um pouco a respeito, isso não é nenhuma novidade. No contexto de desenvolvimento de aplicações, as telas são o ponto de contato entre o sistema e o usuário. E por isso são chamadas de interface. E sim, ela deve ser esteticamente agradável, mas o visual de uma aplicação é somente a superfície de tudo que foi feito para garantir uma boa experiência. No seu livro intitulado *The Elements of User Experience* Jesse James descreve as diversas camadas do processo de projetar uma interface, que vai desde a definição da estratégia até o plano visível ao usuário, a superfície. Em cada plano, existe uma série de decisões a serem tomadas. Cada uma delas influencia as próximas e a experiência final de quem utiliza a interface. Para entender este processo, precisamos visualizar o objetivo de cada camada. ⓘ

• Cada camada pede
• por uma série de
• artefatos, como
• wireframes, user
• needs e diagramas
• de navegação.



Designers devem conseguir ver toda a extensão de um projeto.

Os usuários só conseguem ver a superfície do produto.



5. SUPERFÍCIE

Esta última camada é formada pelas decisões finais a respeito do visual. Imagens, cores, ícones... Como apresentar tudo da melhor forma?

4. ESQUELETO

Logo abaixo da superfície, vem o esqueleto. Uma camada que vai lidar com a localização dos elementos da tela, mas não com sua aparência final.



3. ESTRUTURA

Neste plano, o produto começa a tomar uma forma mais concreta, com definições referentes aos fluxos do usuário e sua navegação pelas telas.

2. ESCOPO

Aqui devem ser definidas todas as *features*, funções e conteúdos do produto. Listar estes elementos da aplicação agora evita surpresas negativas no futuro.

1. ESTRATÉGIA

No plano que deve ser o início de tudo, todos os objetivos do produto, de todos os *stakeholders*, devem ser definidos. O que este produto irá entregar?

DESIGN {

O que os olhos veem, o coração sente!



TEXTO *Tales*
EDIÇÃO *Carolina*
DESIGN *Carolina*
ILUSTRAÇÃO *Carolina*

Tenho certeza que você já ouviu na sua vida frases como essas: “O design desse carro é lindo!” ou “Ela trabalha com design de sobancelhas.”

Menções diversas não vão faltar: design como uma profissão (o ato de fazer design), design como algo já pronto (um produto) ou até mesmo design como sinônimo de criatividade. Mas o que realmente quer dizer a palavra Design? E como ele influencia o nosso dia a dia?

O bom, o belo e o feio

É possível inferir uma tendência de relacionar design com o que é belo, agradável aos olhos. Mas não é só nisso que ele se encontra. Um ambiente pode te deixar ligeiramente desconfortável e isso ser uma boa aplicação dos princípios de design. Algo feio (sim, feio) pode ser uma obra bem sucedida, no sentido de alcançar perfeitamente seu objetivo. Achou esquisito? Sigamos juntos para discutir um pouco disso.

John Heskett, importante figura do Design industrial, afirma que “design, em sua essência, pode ser definido como a capacidade humana de moldar e construir o mundo ao seu redor de formas sem precedentes na natureza, dando sentido à vida e atendendo às suas necessidades”. O design não está preso ao papel, à tela do computador, a uma obra estática: ele se move, é dinâmico e constantemente presente ao nosso

redor. Com essa definição, podemos associar o design a quase todas as profissões. Um engenheiro civil quando faz o desenho de uma casa ou o projeto de um viaduto, o que ele faz? Cria e molda o mundo que o cerca. Quando um carpinteiro faz uma cadeira? Está criando algo que irá satisfazer uma necessidade. Conseguiu acompanhar? Design não é só beleza: também é funcionalidade e a experiência sensorial de quem o usa e o vê.

Somos o que experienciamos

Com isso, toca-se em um ponto que vem sendo discutido bastante hoje em dia: a experiência de usuário ou UX (*User Experience*). Nós, seres humanos, somos seres voltados à experiência. Muitas pesquisas das últimas décadas constataram que experiências trazem mais alegria às pessoas do que posses.

Uma delas foi feita por Thomas Gilovich, professor de psicologia da Cornell University. Em um de seus testes, funcionários foram colocados em uma situação hipotética onde teriam que escolher entre receber um salário alto, mas *menor* do que o de seus colegas ou receber um salário baixo, porém *maior* que o de seus colegas. E muitos dos funcionários não conseguiram se decidir. Por mais que em uma das opções eles recebessem muito mais, o quanto que os outros funcionários receberiam

teve um papel importante nas suas decisões.

Pois quando falamos de posses, é inevitável a comparação com os demais. Um salário alto perde valor e deixa de ser “alto” se outros também o tem.

Então, o teste veio com uma segunda escolha: os funcionários agora poderiam escolher entre ter duas semanas de férias, enquanto seus colegas iriam receber somente uma ou então ter quatro semanas de férias, enquanto seus colegas teriam oito. Mais fácil? Para os entrevistados, sim. As pessoas escolheram quatro semanas praticamente sem hesitação.

Por que isso ocorre? Gilovich afirma que a diferença fundamental é que nós não somos feitos de possessões, mas sim de experiências. Por mais que alguns possam dizer que “o que eu sou está diretamente ligado aos bens que eu possuo”, a palavra “ligado” já entrega a verdade: elas não são parte dele. Já as experiências que tivemos no passado moldaram o nosso “eu”, por isso nos apegamos a elas e as valorizamos tanto. Nós contamos sobre elas em reuniões com os amigos, por exemplo. Imagine o quão chato seria conversar com alguém que apenas fala sobre o que possui.

No experimento Gilovich, as experiências demonstraram ter valor por si só, já o valor das posses dependia do contexto.

Experiência como produto

Uma constatação dessas, para grandes empresas que dependem da venda de produtos, pode parecer assustadora. É necessário se mover para acompanhar essa mudança de paradigma. Devido a isso, as companhias se adaptaram e começaram a vender também a experiência de uso. Estatisticamente, quando temos uma boa primeira impressão de algo, temos a tendência de não notar tantos erros ao longo do tempo, relevando-os. Por exemplo, uma boa embalagem, um material de qualidade, um cartão personalizado, entre outras coisas, são parte da experiência que pode ser entregue junto a um produto.

A teoria Gestalt, no início do século 20, já relacionava a psicologia e a nossa percepção visual. O foco na forma, no todo, afirmando inclusive que “o todo é maior do que a soma de suas partes”. Essa corrente encaixou-se perfeitamente com o design: os produtos/as criações não são apenas um conjunto de atributos, elas são uma entidade completa por si só. Com tudo que vemos, o que não vemos, mas sentimos, e o que pensamos delas quando estamos longe.

Donald Norman, cientista cognitivista, em seu livro “Design Emocional”, explica sobre o conceito das diferentes formas que o nosso corpo experimenta um produto. Mais especificamente, isso acontece em três níveis: visceral, comportamental e reflexivo. Para entender bem, usarei um dos exemplos que ele traz em seu livro.



O Cachorro e a Galinha

Norman usa, para explicar os níveis, galinhas e cachorros (sim).

Pense em uma galinha. Pensou? Elas parecem um pouco burrinhas. Com todo respeito, claro. Mas só pela sua espécie mesmo, nada pessoal. As galinhas têm certas limitações quando comparadas a outros animais na cadeia evolutiva. Ela age por instintos, e não por raciocinar e decidir a melhor atitude para cada momento. Esse *modus operandi* já veio embutido nela. Ao deparar-se com uma cerca de arame com comida do outro lado, a galinha vai ficar batendo no arame, em vão, tentando atravessar o obstáculo, ao invés de dar a volta.

Já um cachorro, não.

Ele pode demorar, mas muitas vezes ele vai descobrir que é possível achar outro jeito de alcançar a comida. E os seres humanos podem ir além. Como trazer a comida para nosso lado para que não precisemos mais dar a volta, por exemplo. Somos seres que pensam no passado, no presente e no futuro. Planejamos e refletimos sobre nossas atitudes (na maioria das vezes).

Com essa situação dá para ilustrar esses três níveis que Norman citou.

A galinha experiencia o ambiente primariamente no nível visceral. Ela analisa o mundo e reage. Apenas reage. Isso está ligado ao seu DNA. Já o cachorro pode operar também no nível comportamental. Ele pode analisar a situação e alterar seu comportamento de

fábrica para então interagir. Comportamento é a palavra chave aqui. E nós humanos, assim como as galinhas e os cachorros, também temos o nível visceral e podemos ver evidências desse processo em nossas emoções subconscientes. Como, por exemplo, quando olhamos para um lugar escuro e sentimos receio de entrar, por associar a algo perigoso. Ou como gostamos de olhar para um céu claro e azul, calmo, por ele indicar que o clima está seguro para ficarmos do lado de fora. À tudo que percebemos, temos uma reação imediata.

E assim como os cachorros, também nos relacionamos com o mundo a nível comportamental. É como nos sentimos quando interagimos com algo. Ele faz bem a sua função? É difícil de usar ou nos frustra por não cumprir bem seu objetivo? É nesse nível dos processos mentais que estão as nossas tarefas diárias.

O nível visceral se comunica com nossos sentimentos “de fábrica” já o comportamental está associado às nossas ações





E por fim, o que nos difere dos outros seres vivos, é a nossa capacidade reflexiva. Podemos pensar nas experiências passadas e antecipar as futuras. Justamente por isso temos uma forma a mais de experienciar o mundo.

Um exemplo disso, aplicado ao design, é quando falamos de um objeto querido por nós, pode ser um amuleto da sorte qualquer, uma pedrinha até. Parece ser algo inútil, feio, mas que traz um forte apego emocional, uma conexão. A relação com uma obra de arte é semelhante. Ao observarmos um quadro, ele pode ser uma imagem que inicialmente nos traz repulsa, mas com contemplação e conhecimento do contexto histórico e social da peça, percebemos a intenção do artista e “sobrescrevemos” essa sensação e impressões iniciais a respeito da obra.

Um bom design precisa conseguir sucesso nesses três níveis. Mesmo que o sucesso seja dependente da situação específica e de cada ser humano. Nós refletimos em experiências passadas para formar o nosso eu, lembra? Tais experiências são certamente diferentes, sendo influenciadas pela cultura e ambiente que estamos inseridos.

Design Salva-vidas

Vamos realizar um exercício de reflexão: pense no design de uma sala de controle de uma usina nuclear. Quando tudo está calmo, a sala está agradável, podendo até ter uma música bem baixa de fundo. Tudo isso para que ela seja um ambiente minimamente relaxante. Relaxamento esse que também nos favorece ter boas ideias, pois nos momentos

leves, visceralmente falando.

No entanto, na hora que ocorre algum problema a atmosfera da sala precisa se adaptar: as luzes mudam, a música para, sirenes começam a tocar. Isso serve para ligar o nosso “sinal de alerta” no cérebro. A resposta do nosso corpo para tais estímulos é a tensão, gerada intencionalmente pelos alertas. O nosso cérebro se foca: é necessário resolver o problema e remover o incômodo. Contudo, se uma situação for adversa demais, o indivíduo pode acabar tendo uma visão tunelada: ver o problema e nada além dele. Um exemplo é que ao estar fugindo de um incêndio, muitas pessoas empurram as portas, apesar de esta atitude cerrar a porta em alguns casos, invés de abri-las. Isso já causou tantos problemas no passado que agora existe a barra antipânico, sendo obrigatória em vários locais, nas saídas de emergência, para que você apenas empurre e a porta se abra. Olha o design salvando vidas. Do mesmo modo, ao se criar o design de interações virtuais, é importante atentar para essas reações automáticas e associações naturais que possuímos. Por exemplo, fazer com que quando o aplicativo necessite de foco e atenção extrema do usuário, emita sons mais estridentes e utilize cores como o vermelho.

O Design vai além do que é belo. Muito além de um adereço, um *frufu*. É um conceito que nos acompanha desde sempre, que pode trazer melhorias e mudanças significativas na vida das pessoas e que tem um papel importante na construção das nossas memórias e experiências. ❶

CÓDIGO {

Por que *não* utilizar constraints de left and right?

TEXTO Karolina
EDIÇÃO Brena
DESIGN Carolina
ILUSTRAÇÃO José João

Saudações, caros leitores deste humilde artigo. Todos sabemos que é um pouco difícil escrever um artigo. E, assim como um artigo, as *constraints* exigem certos conceitos um pouco confusos, que no final tornam um momento - que deveria ser agradável - em pesadelo. Mas calma, que este artigo tem por objetivo auxiliar na sua tão temida configuração de *auto layout*, mais especificamente nas diferenças entre as âncoras *trailing*, *leading*, *right* e *left*. Além disso, teremos uma breve aplicação em um trecho de código.

Uma imagem vale mais do que mil palavras. Mas dessa vez não. As âncoras *Center X* e *Center Y* sempre dizem respeito ao centro de uma *view*. Nessa vertente, você irá notar que a sua *view* ficará num centro horizontal e vertical, definindo essas duas *constraints*. Existem outras definições do *NSLayoutAnchor* para aqueles que gostam de aprofundar os conteúdos, mas por enquanto vamos nos limitar ao “básico”.

Quando queremos que uma *view* fique em um ponto específico da tela, ao lado de outra *view*, por exemplo, é normal pensar em *constraints* de *left* e *right*. É uma boa decisão se você não vai levar sua aplicação ao *Testflight*. Quando tratamos de escrita e línguas, é necessário compreender que nem todas as regiões tem o mesmo padrão. Nós estamos num país em que fazemos a leitura de um artigo da esquerda para a direita, e de cima para baixo. É algo tão intrínseco, que muitas vezes

enfrentamos problemas ao ler mangás, que por serem quadrinhos orientais possuem a leitura da direta para a esquerda. Quando nos referimos às *constraints left* e *right*, nós somente estamos pensando num contexto limitado ao fluxo de leitura ou escrita que surge da esquerda, e finaliza no lado direito. Mas o que acontece quando lidamos com pessoas que utilizam o sistema de escrita oposta?

É incômodo para o usuário ver a sua própria linguagem ao contrário, sendo obrigado não somente a se adaptar às tecnologias, mas ainda ver seu costume ser simplesmente ignorado. Para isso, surgiram as *constraints* de *leading* e *trailing*.

O *leading* é relativo ao início do sentido de leitura, enquanto que o *trailing* é relativo ao final. Importante denotar que ao utilizar uma ou ambas as *constraints* nós estamos com a *NSLayoutXAxisAnchor*, então elas podem ser combinadas em relação à outras propriedades desta classe. Entretanto, não deve-se misturar *trailing* e *leading* com *left* e *right*, uma vez que são atributos que possuem funcionamento diferente.

Suponha que queremos inserir uma *UILabel* alinhada à esquerda em nossa *ViewController*:

Perceba que ao definir a *constraint* de *leading* a nossa *UILabel* estará posicionada do lado esquerdo da tela - quando a linguagem do dispositivo é Português.

Mas se mudarmos a linguagem do dispositivo pra Árabe, por exemplo, que é uma linguagem RTL

```
class ViewController: UIViewController {  
  
    let titleLabel: UILabel = {  
        let label = UILabel()  
        label.font = .boldSystemFont(ofSize: 24)  
        label.text = "Hello!"  
  
        label.translatesAutoresizingMaskIntoConstraints =  
false  
        return label  
    }()  
  
    override func viewDidLoad() {  
  
        ...  
  
        view.addSubview(titleLabel)  
  
        NSLayoutConstraint.activate([  
            titleLabel.topAnchor  
                .constraint(  
                    equalTo: view.topAnchor,  
                    constant: 100),  
            titleLabel.leadingAnchor  
                .constraint(  
                    equalTo: view.leadingAnchor,  
                    constant: 16),  
            titleLabel.trailingAnchor  
                .constraint(  
                    equalTo: view.trailingAnchor,  
                    constant: -16)  
        ])  
    }  
}
```

(Right to Left), o dispositivo reposiciona automaticamente a *UILabel*.

Com esta dica em mente, espero que você tenha momentos mais agradáveis configurando o *Auto Layout*! 📌

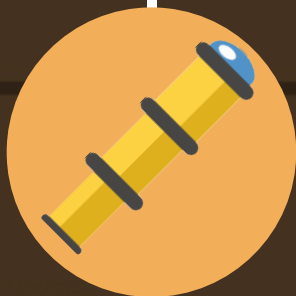
INFOGRÁFICO {

Fluxo de desenvolvimento com Gitflow

TEXTO José João
EDIÇÃO Anderson
DESIGN José João
ILUSTRAÇÃO José João

O *Gitflow* parte de que devemos ter duas *branchs* principais: a *branch main*, que será sempre a última versão lançada para usuários e a *branch developer* que conterá as *features* que estão compondo a nossa versão do projeto.

TUDO COMEÇA DO INÍCIO



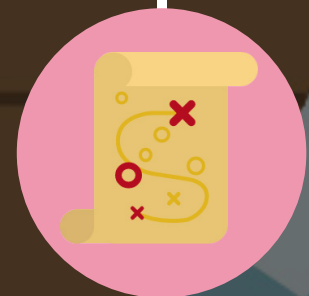
AQUI NÓS CRESCEMOS

A *branch developer* terá o conjunto de todas as novas *features* que estão sendo desenvolvidas para uma nova versão do projeto. É de extrema importância que todos os desenvolvedores mantenham sua *branch developer* local sempre atualizada, dessa maneira conseguimos garantir que nenhuma *feature* será perdida.



E é na *branch de feature* que cada funcionalidade é desenvolvida. A ideia é que após o seu desenvolvimento, ela seja integrada com a *branch developer*. Dessa maneira, a equipe consegue ser dividida em várias *branchs* menores de *feature*, facilitando o desenvolvimento paralelo. Por padrão, usamos o prefixo **feature/** para o nome da *branch*.

DE FEATURE EM FEATURE...





Durante o desenvolvimento, é comum que apereçam bugs. Para isso, é designada a uma *branch* específica realizar a reparação de maneira rápida e discreta, chamada de HOTFIX. Geralmente é criada para erros direto em master, então são *branches* de rápido e pequeno desenvolvimento.



NÃO ACERTAMOS SEMPRE

A GENTE SE VÊ POR AQUI

Após o desenvolvimento de algumas *features*, antes da data de lançamento, é essencial criar a *branch* de release: ela será a *branch* responsável para ser uma espécie de BETA, em que serão testadas e consertadas todas as *features* feitas.



E ESTAMOS AQUI

No fim, definimos um fluxo baseado em entregáveis menores e com um tempo limite de desenvolvimento. Vale salientar que o Gitflow pode ser adaptado para a realidade da equipe para que haja uma maximização dos seus resultados. **1**

DESIGN {

Gameificação para apps

TEXTO *Mateus N.*
EDIÇÃO *Carolina*
DESIGN *David*
ILUSTRAÇÃO *David*



O termo gameificação (ou *gamification*) já não é mais um termo tão novo para a maioria das pessoas. Ele consiste no processo de aplicar conceitos de jogos em coisas que não são jogos com a finalidade de aumentar a motivação e o engajamento do usuário.

Empresas, escolas e outras instituições já reconhecem os benefícios da gamificação e a estão aplicando para melhorar os seus processos e resultados. Muitos aplicativos também já fazem uso desse recurso com muito sucesso para melhorar a experiência do usuário.

O que muita gente não sabe é que gamificação é muito mais (e deve ser) do que apenas dar pontos, emblemas, medalhas e criar rankings de usuário. Nesse artigo vamos ver que essas estratégias correspondem apenas a uma pequena parte da gamificação e vamos ver na prática diversas outras estratégias sendo aplicadas em aplicações mobile.

Motivação

Em todas as nossas atividades precisamos de algo para nos mover, algo para dar sentido ao nosso gasto de energia e optarmos por fazer algo que até não gostaríamos de fazer. A esse impulso damos o nome de motivação. É ela que faz um trabalhador acordar 6:00h de uma manhã de segunda mesmo estando com sono e preferindo ficar em casa assistindo série, por exemplo. Ela pode vir através vários objetivos, seja pra conseguir dinheiro no fim do mês, seja pra honrar um compromisso ou simplesmente por diversão. E a motivação está extremamente relacionada com a gamificação, pois é pensando em aumentar a motivação de determinada pessoa para determinada

atividade que fazemos o uso de gamificação.

Costuma-se dividir a motivação humana em duas grandes áreas:

Motivação Extrínseca: Motivação causada através de meios externos ao usuário. Ela pode ser vista como um motivação por receber algo de alguém, sendo alcançada através de: premiações, promoções, vantagens e etc.

Motivação Intrínseca: Motivação causada por fatores intrínsecos ao usuário. Como a sensação de fazer parte de algo maior, ao alcance de metas pessoais, autorrealização, autoestima e etc. Essa motivação tende a ser muito mais eficiente do que a primeira, pois trata-se de algo interno da pessoa e é muito mais difícil de ser removida (em contrapartida, também é mais difícil de ser criada, mas isso não deve ser um empecilho).

O modelo Octalysis

O modelo Octalysis é um modelo criado por Yu-Ki-Chou, uma das maiores autoridades do mundo em gamificação, que divide as estratégias de motivação em 8 categorias: significado, empoderamento, influência social, imprevisibilidade, evitação, escassez, propriedade e conquistas.

Eu resolvi então escolher alguns aplicativos que normalmente utilizo e que fazem uso desses elementos para explicar cada um deles nesse artigo. Obviamente cada aplicativo pode fazer uso de elementos de mais de uma categoria, mas o objetivo era expor uma quantidade maior de aplicativos para vermos mais exemplos aplicados.

Significado: Essa categoria diz respeito ao senso que o jogador tem de fazer parte de algo maior do que ele mesmo. O ser humano, de modo geral, é apaixonado por relacionamentos. Nós gostamos de saber que não estamos só.

Conquistas: O fato de ser possível que o jogador conquiste coisas dentro do jogo através do seu esforço. Aqui estão alguns dos elementos de gamificação mais conhecidos, como a capacidade de ganhar pontos e outras conquistas.

Empoderamento: A capacidade que o jogador tem de conseguir poder de alguma forma.

Influência social: A capacidade que o jogador tem de exercer influência sobre outros jogadores

Evitação: Elementos no jogo ou sistema que os jogadores têm que ou preferem evitar.

Imprevisibilidade: A capacidade que o jogo ou sistema tem de gerar condições aleatórias e imprevisíveis para os jogadores.

Escassez: O fato de alguns recursos dentro do jogo ou sistema serem limitados e quem nem todos podem ter em abundância. O check de verificado no instagram.

Propriedade: Um senso de pertencimento sobre algo que o usuário tem. Pode ser sobre algum item, moedas, ou dinheiro do jogo.

Vimos nesse artigo uma série de elementos motivadores que poderemos utilizar dentro da nossa solução gamificada. Claro que isso tudo é só uma pincelada sobre o tema e não adianta simplesmente pegarmos as características desses elementos e irmos aplicando sem uma análise prévia. É muito importante conhecermos o nosso público-alvo e avaliarmos os principais tipos de jogadores dentro desse público. Só assim conseguiremos construir uma solução personalizada e realmente motivadora para eles. ①



Desbravando games para Macs

TEXTO *Magno*
EDIÇÃO *Brena*
DESIGN *David*
ILUSTRAÇÃO *David*



Swift Student Challenge

Antes de desbravar esse mundo de games pra Macs com vocês, gostaria de contar um pouco da minha experiência sobre como comecei com Sprite Kit.

Conheci o Sprite Kit durante o Swift Student Challenge na WWDC 2020. Em que tive apenas uma semana pra concluir meu projeto. Logo quando comecei a pesquisar sobre o que queria fazer, jogos foram a primeira coisa que veio à minha mente, porque sempre quis aprender a criar jogos e pensei que seria a oportunidade perfeita para isso.

Estava bastante empolgado, mas percebi que fazer um jogo envolvia muito mais do que eu sabia na época. Envolvia assuntos como Design para jogos e eu não tinha muito conhecimento sobre isso. Então, a primeira coisa que fiz antes de começar, foi aprender a fazer o design de um game.

Design dentro da temática de games é um tópico bastante único, porque difere do de interfaces pra dispositivos móveis que segue guias como a HIG (Human Interface Guidelines), não segue padrões e seu objetivo é ser único para cada game criado. disclaimer: Algumas ideias gerais, como cores e tipografia podem ser aplicadas assim como aprendemos no design de interfaces.

Sabendo que design não era meu forte, decidi focar mais nas mecânicas para tornar o game mais simples. Depois de terminar meu projeto, decidi compilar alguns resources que encontrei no meio do

caminho sobre a criação de jogos para compartilhar com a comunidade do Academy.

Antes de começar seu jogo, é importante se organizar principalmente se você tiver um prazo. Uma maneira interessante de fazer isso é por meio do Game Design Document (GDD) pode ser como um CBL feito exclusivamente pra criação de jogos. Nele, você pode documentar e definir seus objetivos como temática, mecânicas, plataformas, inimigos, levels, entre outros.

Tecnologias Apple

Existem muitas tecnologias que a Apple disponibiliza para o game developer. E é importante conhecer as *features* que cada framework oferece para que você possa aplicá-lo. Como o GameKit, por exemplo, que permite que os jogadores interajam com amigos e participem de jogos multijogador. Ou o SceneKit que possui recursos para criar cenas e efeitos animados em 3D. Eu queria utilizar a mecânica jogador x computador, mas só é possível fazer isso com o GameKit que está mais relacionado à IA, enquanto eu pretendia utilizar apenas o SpriteKit. Então, decidi deixar essa funcionalidade de lado por enquanto e focar no SpriteKit.

Uma maneira fácil de entender a capacidade do

SPRITEKIT É UM FRAMEWORK USADO PARA CRIAR JOGOS 2D DE ALTO DESEMPENHO

SpriteKit, é olhando em geral como ele funciona. Segundo a documentação da Apple, esse framework é usado para criar jogos 2D de alto desempenho e baixo consumo de energia. Possui integração com SceneKit, efeitos físicos e animações avançadas. Permite adicionar campos de força, detectar colisões e gerar efeitos de iluminação em jogos.

Um jogo trabalha bastante com o que eu posso chamar de While True, em que ele sempre estará (de forma controlada) pegando atualizações e interações do usuário com a tela. Por conta disso, ele pode consumir um pouco mais das máquinas. Para isso, o SpriteKit e o Xcode, em conjunto, otimizaram o consumo de energia melhorando o desempenho. Logo, tentar fazer um jogo com outras ferramentas que não têm a mesma preocupação, como Appkit e UIKit, talvez torne o jogo um pouco mais pesado para o usuário se, assim como eu, você ainda não saiba como lidar com isso.

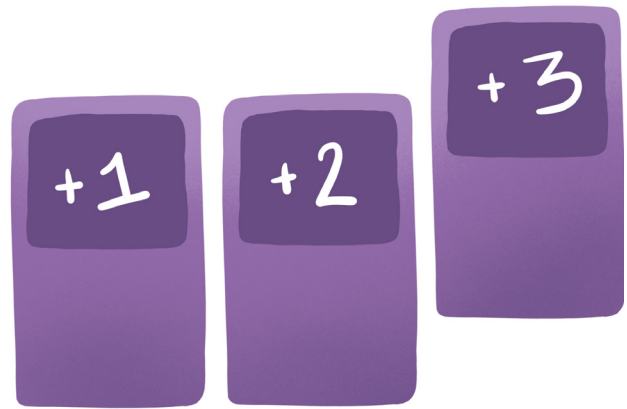


Sabendo de tudo isso, como eu faria para criar um jogo com tal poder?

Caso já tenha usado o UIKit, você percebeu como essa ferramenta possui diversos componentes prontos para serem posicionados e utilizados da maneira mais interessante para seu aplicativo. Com o SpriteKit é semelhante, porém com um foco especial em mecânicas, tais quais ações e física que foram as que eu mais aprendi e apliquei no escopo do meu jogo.

No SpriteKit, ações são definidas pelas SKActions, que possuem diversos modos diferentes de interações com os objetos e cenário, podendo mover, animar, rotacionar, além de diversas outras possibilidades. Caso você pretenda fazer um jogo em que a dinâmica do jogo está em detalhes mais simples como: uma animação aqui, ou clicar com o mouse e o personagem se mover, alguma micro iteração aqui e acolá. Aplicar SKActions pode ser deveras interessante para o projeto, uma vez que elas são bem efetivas nisso.

Caso o que você pense para o jogo que quer cons-



truir seja algo mais complexo e que vá exigir leis físicas que rejam como esse mundo funciona, a dica é manter o foco no SKPhysics e seus adjacentes. É interessante pensar em cada detalhe e utilizar áreas em uma cena que funcionam de acordo com o SKPhysics que você criou, pois caso você pretenda criar diferentes lugares com diferentes leis, não vai ter tantos problemas quanto eu tive quando utilizei pela primeira vez. Nisso, a matemática (e a própria física) pode ser a chave para poder alcançar o resultado esperado. No jogo em que criei, eu utilizei apenas de SKActions, porque percebi que o que eu pretendia criar não envolvia física ou qualquer dinâmica complexa entre os elementos da tela, tais quais colisão ou repulsão, pois se tratava de um simples jogo de cartas com elementos estáticos, logo não havia necessidade de aplicar SKPhysics e nem seria inteligente, sabendo que as SKActions já supriam todas necessidades do meu jogo.

Um detalhe muito importante e que pode ajudar bastante em como irá criar o seu jogo com SpriteKit, é saber como funciona a sua arquitetura. Que não é algo tão complexo e vai ajudar bastante. O SpriteKit utiliza bastante do que chamamos de Nodes (SKNodes). Todos os componentes acabam sendo parte deles, além disso, a maioria dos componentes são customizáveis e criados pelo próprio desenvolvedor (eles também são Nodes). Interessante salientar que eles são capazes de possuir outros Nodes como seus filhos, e que fazem parte deles.

Sabendo disso será muito mais fácil pensar em soluções e detalhes que você deseja adicionar ao seu jogo, e também em como implementá-los.

Sabendo disso se tornará muito mais fácil pensar em soluções e detalhes que deseja adicionar ao seu jogo, e também em como implementá-los. ❶

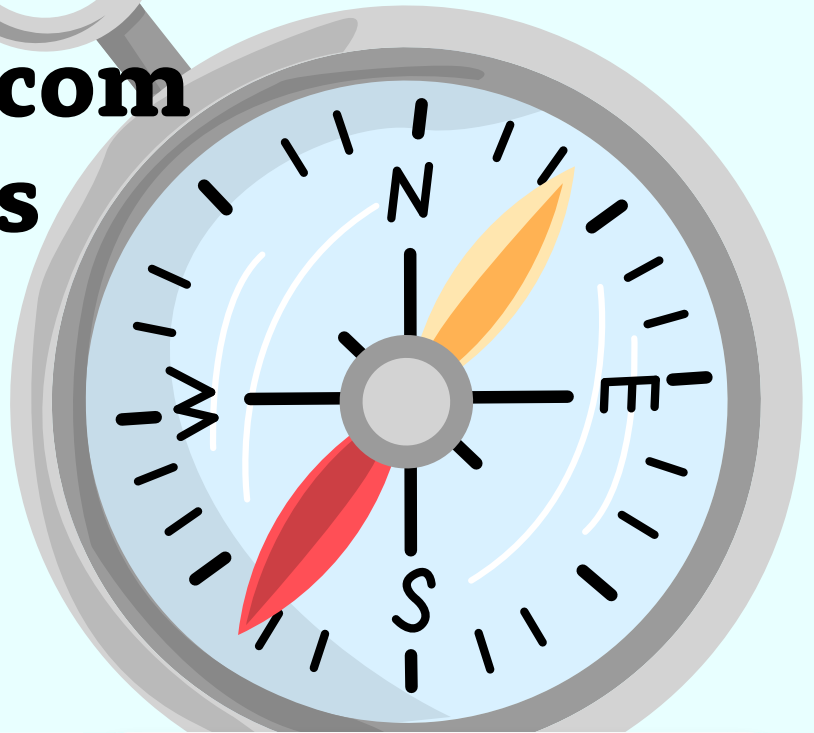


Use o QRCode para ver o resultado do meu projeto do Playgrounds ou acesse youtu.be/dWmJKZeuMew

CÓDIGO {

Navegando com coordinators

TEXTO Vinícius
EDIÇÃO Beatriz
DESIGN Carolina
ILUSTRAÇÃO João



Implementar a maneira tradicional de centralizar toda lógica de funcionamento em uma *View-Controller* é dizer não a uma aplicação escalável. É se afastar da ideia de todas as aplicações que desejam se manter no mercado, receber atualizações e serem mantidas.

É por isso que muitos padrões de projetos foram criados para suprir essa necessidade, *coordinator* é um exemplo disso, sua primeira aparição foi no livro “*patterns of enterprise application architecture*” em Novembro de 2002 e se popularizou entre a comunidade iOS em 2015 com a apresentação de Saroush Khanlou na NSSpain Conference que logo após escreveu diversos artigos sobre o assunto.

Coordinator ou *Director* é um padrão de projeto que objetiva desacoplar responsabilidades de fluxo de navegação removendo isso de suas controllers, você será capaz de controlar todas as rotas da sua aplicação.

Por exemplo, digamos que você queira fazer um fluxo muito comum de uma *TableView* para uma tela onde irá mostrar os dados com mais detalhes.

Apesar de ser uma forma bem comum de implementação, existem diversos problemas relacionados a essa abordagem, conforme os fluxos de tela crescem dentro da aplicação, você poderá ter várias chamadas da mesma *DetailsViewController*. →



```
override func tableView(_
tableView: UITableView,
didSelectRowAt indexPath:
IndexPath) {
    let detail =
dataSource[indexPath.item]
    let destinationController
= DetailViewController()

destinationController.detail =
detail

navigationController?.pushViewController(destinationController,
animated: true)
}
```

Você também deve ter percebido que é impossível utilizar esse mesmo fluxo para uma mesma situação de chamada dessa mesma controller, ou seja, você precisaria copiar e colar esse código todo novamente.

Além disso, a *DestinationController(child)* está executando uma ação diretamente na *navigationController(parent)*, o que pode oca-

sionar diversos problemas caso você não controle da sua árvore de navegação. Mas podemos evitar isso ao utilizarmos *Coordinators*.

Use Coordinators!

A comunicação entre controllers será feita apenas através de coordinators, os resultados que você irá encontrar são diversos, primeiramente todas os fluxos serão reusáveis, você não precisará codificar fluxo de navegação caso você utilize a mesma tela em diferentes propósitos.

Para implementar coordinator na sua aplicação, primeiramente é necessário implementar uma classe geral responsável pelas rotas gerais da sua aplicação. Todo classe coordinator possui um

array com seus próprios *Childs Coordinators*.

AppCoordinator

O Application coordinator é a classe que irá dar início ao fluxo das coordinators e também será responsável por inicializar novos coordinators.

Depois disso (child)coordinators podem ser criadas para tasks específicas de login ou dados de profile. cada coordinator é inicializado pelo (parent)coordinator.

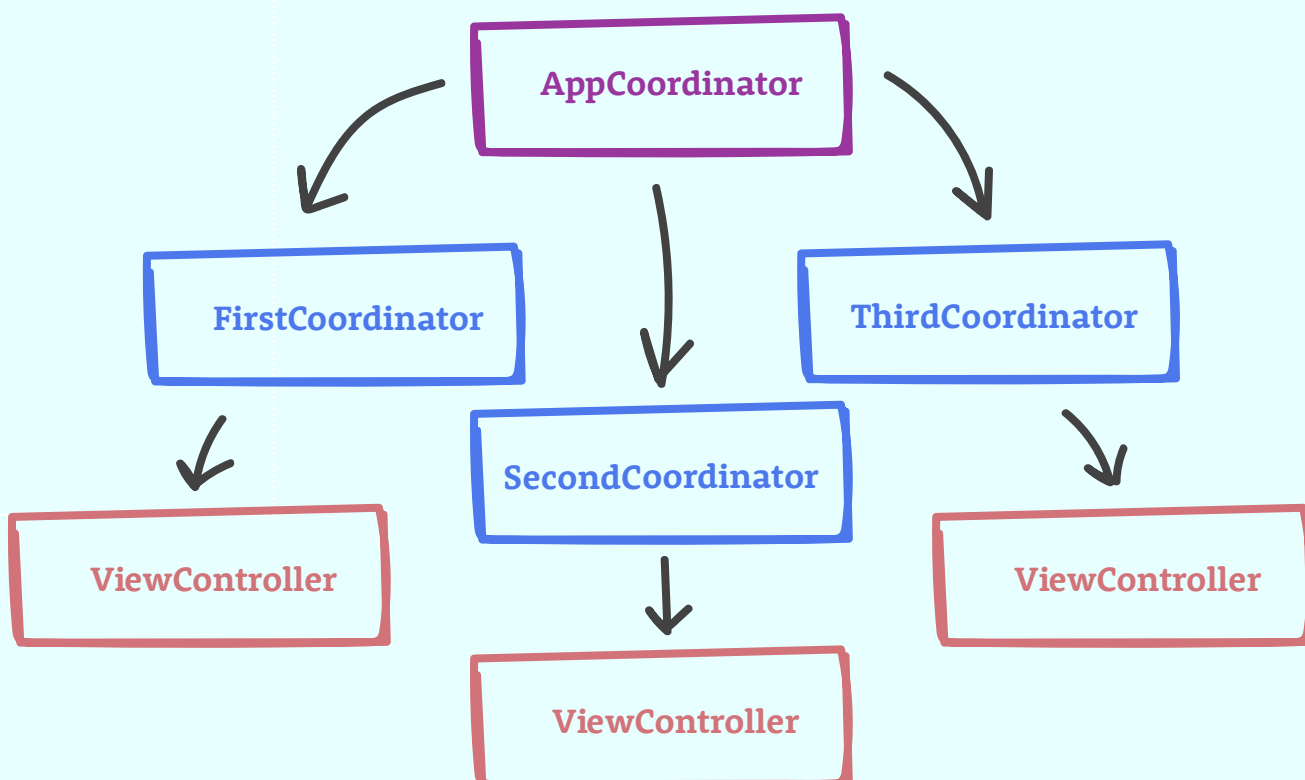
Um coordinator, assim como qualquer outro objeto, para existir em memória precisa pertencer a outro objeto, Geralmente, quando a navegação do sistema é criada por coordinators, alguns deles criam outros e

armazenam uma referência dentro do objeto, depois do coordinator filho finalizar, a memória deve ser liberada.

Protocol e clousures

Muitas pessoas podem acabar confundindo coordinators com delegates, mas a verdade é que coordinators são delegates que possuem a função específica de navegação. Já que, delegates basicamente permitem você injetar dependências para outras entidades, mas para nós, não faz sentido uma outra controller ter essa responsabilidade, então delegamos essa função ao nosso coordinator, afinal é ele que deve saber para onde temos que ir.

Isso é como um coordinator se parece. →



```

    ● ● ●

protocol Coordinator {
    // Inicia a rota do coordinator
    func start()
    // Escolhe qual coordinator deve ser iniciado
    func coordinate(to coordinator: Coordinator)
}

extension Coordinator {
    func coordinate(to coordinator: Coordinator) {
        coordinator.start()
    }
}

```

```

    ● ●

final class ObjectListCoordinator: Coordinator {

    var navigationController: UINavigationController!

    init(navigationController: UINavigationController) {
        self.navigationController = navigationController
    }
    func start() {
        let objectListViewController = ObjectListViewController()
        navigationController?.pushViewController(objectListViewController, animated: false)
    }
    // Função que envia para a tela de detalhes.
    func showDetails(object: ObjectDetailsViewModel) {
        let objectDetailsViewController = ObjectDetailsViewController(coordinator: self, viewModel:
object)
        navigationController?.present(objectDetailsViewController, animated: true, completion: nil)
    }
}

```

```

    ● ● ●

class ObjectListViewController: UIViewController {

    let viewModel = ObjectListViewModel()
    var coordinator: ObjectListCoordinator?

    override func viewDidLoad() {
        super.viewDidLoad()
        // Inicializa a instância do coordinator.
        coordinator = ObjectListCoordinator(navigationController: navigationController!)
    }

    override func tableView(_ tableView: UITableView, didSelectRowAt indexPath:
IndexPath) {
        // Envia os dados e inicia a ObjectDetailsViewController
        coordinator?.showDetails(object: viewModel.getDetail(for: indexPath.item!))
    }

}

```

```

    ● ●

class ObjectDetailsViewController: UIViewController {

    let viewModel: ObjectDetailsViewModel
    let coordinator: ObjectListCoordinator

    init(coordinator: ObjectListCoordinator, viewModel: ObjectDetailsViewModel) {
        self.coordinator = coordinator
        self.viewModel = viewModel
    }
}

```

Tudo fica melhor com Coordinators.

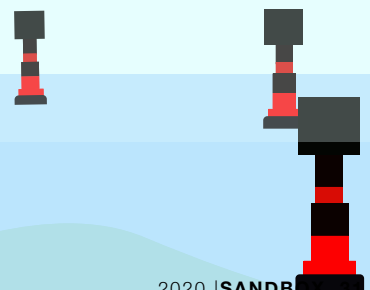
Com a implementação de coordinators, é possível até mesmo criar um flow inteiro de aplicação para realizar testes A/B.

Então, se você está embarcando em um grande projeto, lembre-se de navegar com Coordinators!

Assim você poderá remover toda logica de navegação da controller, ter um código limpo, reusável, manutenível e fácil de ser testado e poupar muito trabalho e dores de cabeça no seu processo de desenvolvimento. 📌



```
class Coordinator_Tests: XCTestCase {  
  
    func  
    test_when_onNext_viewModelDetail_then_flow_showD  
etail() {  
  
        // Dado  
        let navigationController =  
        UINavigationController()  
        let viewModelFactory =  
        StubViewModelFactory()  
  
        let coordinatorFlow =  
        SpyCoordinatorFlow(  
            navigationController:  
            navigationController,  
            viewModelFactory: viewModelFactory)  
  
        let coordinator = MainCoordinator(  
            navigationController:  
            navigationController,  
            coordinatorFlow: coordinatorFlow,  
            viewModelFactory: viewModelFactory)  
  
        coordinator.start()  
  
        // Quando  
        viewModelFactory.recordedMasterViewModel?.detail.  
onNext(()  
  
        // Então  
        XCTAssertEqual(coordinatorFlow.showDetailCalls,  
        1)  
    }  
}
```



HISTÓRIAS {

From Coder To Devigner

OS ALUMNI DO ACADEMY IFCE LEVY CRISTIAN E AMANDA TAVARES
COMENTAM SOBRE O SUAS EXPERIÊNCIAS ATUANDO COMO DEVIGNERS.

TEXTO *Fernando*

EDIÇÃO *Brena*

DESIGN *Fernando*

Devigner é um termo popular usado para definir profissionais que possuem conhecimento em programação e design.

Entrevistamos dois ex-alunos do Academy IFCE que possuem skills em ambas as áreas para saber melhor como foi a experiência de ingressar no academy como coder e acabar se tornando um devigner. A primeira entrevistada, Amanda Tavares trabalha como UX Designer no Instituto Atlântico e como trainee de design no Academy IFCE. Já Levy Cristian trabalha como Engenheiro de Software no Instituto Atlântico. Tanto Amanda quanto Levy já tiveram experiência no mercado de trabalho como desenvolvedor(a) e como designer, e irão responder algumas perguntas sobre como é ser um devigner.

O Devigner é um profissional que possui uma base sólida de conhecimentos de design e também é um ótimo coder. Você se considera um Devigner?

Amanda: No início do Academy, eu me considerava uma programadora que conhecia um pouco de design. Quando entrei no Academy comecei a me sentir verdadeiramente uma coder, porém, neste mesmo período foi surgindo meu interesse por design porque comecei a ver o design como algo além de telas bonitas. No segundo ano

de Academy, comecei a estudar User Experience e fundei, junto com o Vincenzo - um dos mentores de design da turma 2018/2019 -, o grupo de estudo de UX do Academy, onde começamos a ver muita coisa legal e foi quando comecei a considerar UX design como carreira. Hoje, após um ano da minha formatura, eu já tive uma boa experiência com código e estou estudando bastante design, então, eu me considero sim Devigner.

Levy: No primeiro ano do Academy, eu me sentia um coder que passou a aprender um pouco mais sobre design. Comecei inspirado principalmente pelo processo de branding e me apoiava muito nos mentores de design. Porém, chegou um momento em que eu comecei a debater com os mentores sobre o design propriamente dito. Como eu já trabalhei tanto como coder quanto como designer, hoje em dia eu me considero devigner.

Teve algum momento específico em que você parou e refletiu que você poderia ser tanto coder quanto designer?

Amanda: Eu estudava muito sobre interfaces porém quando passei a estudar as teorias de UX de Jesse James, percebi que as etapas que eu mais gosto de fazer são estratégia e escopo.



Amanda Tavares
UX Designer

Comecei a pesquisar em como isso funciona no mercado e eu acabei vendo que existem diversos profissionais de User Experience, então porque não testar? Comecei então a estudar mais a área e me interessar cada vez mais.

Levy: No início do segundo ano, eu percebi que estava com um bom background tanto de design quanto de código, então eu vi que realmente conseguiria tanto ser coder quanto designer nos desafios.

No conhecimento empírico, design é algo subjetivo. Nada além de coisas bonitas mas, conforme nos aprofundamos na área, percebemos que não é apenas isso. Quando você ingressou no Academy também tinha essa visão?

Amanda: Eu também acreditava nisso. Achava design muito interessante, mas não queria me aprofundar naquilo porque acreditava que design era muito subjetivo. Acreditava que o design era um dom, que eu precisaria nascer de novo para ser boa naquilo. Nessa perspectiva, o design não era algo que me atraía, porém, quando, no Academy, eu compreendi que o design é tudo desde a estratégia à superfície, percebi que design era para mim.

Levy: Quando entrei no Academy eu achei que design era só telinhas bonitas mesmo. Achava que do nada as telas iriam surgir, mas minha visão mudou, e eu percebi que o design é um processo, onde você tem inputs que te geram um resultado e esse resultado vai ser algo que ainda não existe por padrão e que vai solucionar algum problema.

Essa visão de que o design é somente sobre coisas bonitas na tela é o que faz com que o programador tenha dificuldade em se interessar em estudar design?

Amanda: Eu acredito que sim.

Levy: Eu acredito que sim,

“O NÃO-CONCEITO CRIA NOS CODERS A VISÃO DE QUE DESIGN É SÓ CRIAR TELINHAS, FAZENDO COM QUE OS MESMOS DESVALORIZEM O PROCESSO DE DESIGN.”

a falta de conhecimento sobre algo cria um preconceito, um não-conceito, por parte dos coders. Isso cria neles uma visão de que o design é só criar telinhas fazendo com que essas pessoas desvalorizem o designer e o processo de design.

Quando você inicia no Academy IFCE é comum que você tenha uma base de programação e pouco ou nenhum conhecimento de design. Isso pode fazer com que os alunos, mesmo que se interessem por design, tenham receio de investir na área por terem que começar do zero. Como você lidou com isso?

Amanda: Eu acho que sempre rola muito um sentimento de que para ser um bom designer, eu preciso de mais do que eu já sei. Na verdade, esse sentimento é presente em todas as áreas, você sempre vai sentir que não sabe tudo. O conhecimento é uma fonte inesgotável, você nunca vai saber sobre tudo e isso, particularmente, me conforta. Todas as experiências e conhecimentos que você absorve são vantagens e mesmo que você nunca saiba tudo sobre o design, o que você sabe sempre vai ser um diferencial.

Levy: Eu tinha esse receio no começo, porém eu presenciei um momento no Academy, onde eu escutei que: “Nunca é perda de tempo quando estamos falando de conhecimento, conhecimento só engrandece a pessoa”. Naquele momento eu aprendi que, na verdade, eu estava mentindo para mim mesmo para que eu não estudasse design e continuasse na minha bolha. Design está lá para você aprender, para isso, você precisa sair da sua zona de conforto.

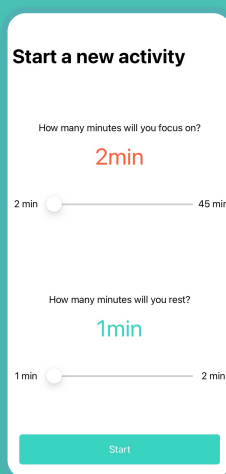


Levy Christian

Eng de Software

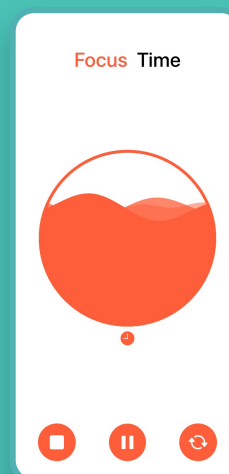
“NUNCA É PERCA DE TEMPO QUANDO ESTAMOS FALANDO DE CONHECIMENTO, CONHECIMENTO SÓ ENGRANDECE A PESSOA”

“TODAS AS EXPERIÊNCIAS E CONHECIMENTOS QUE VOCÊ ABSORVE SÃO VANTAGENS”



Tempo

Take your time



Como você está atualmente ativo no mercado de trabalho, pela tua experiência, como o mercado de trabalho enxerga os devigners?

Amanda: Quando eu sai do Academy busquei inicialmente oportunidades de emprego como desenvolvedora. Sempre que eu apresentava meu background de design, os entrevistadores ficavam impressionados e isso

foi um diferencial para conseguir a vaga de desenvolvedora. E recentemente eu consegui uma vaga como designer onde ser desenvolvedora foi um diferencial. Nessas entrevistas, eu sempre demonstrava o quanto todas as minhas experiências e conhecimentos poderiam contribuir para a empresa.

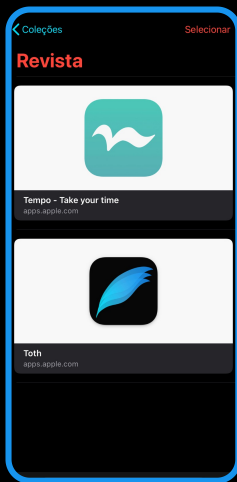
Levy: Dependendo da

empresa, o devigner pode ser muito bem visto ou não. Existem empresas que preferem especialistas e que precisam de especialistas. O devigner é algo novo, algo diferente, isso pode fazer que a empresa ache que a pessoa sabe pouco porque sabe os dois. Porém, eu já estive em empresas em que eu pude trabalhar em ambas as áreas.

E quando você não sabe se quer trabalhar agora como coder ou como designer?

Amanda: Eu acredito que a gente tem a visão de que você precisa agora, nos seus 20 e poucos anos, decidir o que você quer para o resto da sua vida. Você pode hoje decidir ser designer trabalhar com aquilo

“O DEVIGNER É ALGO NOVO, ALGO DIFERENTE, ISSO PODE FAZER QUE A EMPRESA ACHA QUE A PESSOA SABE POUCO PORQUE SABE OS DOIS.”



Toth

Do more with less



por um tempo ou como coder e somente mais tarde decidir qual dos dois você quer se especializar. Eu acredito que a gente precisa focar no presente, o que você quer hoje?

Levy: Eu passei por este momento de dúvida no final do Academy e no meu caso eu percebi que eu sou apaixonado por criar e chega um momento, como designer, que você para

no projeto e eu queria ir além. Ainda amo os dois mas, atualmente, eu prefiro codar justamente pelo feeling de criação.

Durante o Academy, como você balanceou o estudo de ambas as áreas?

Amanda: Eu sempre foquei em não pular etapas, em todo challenge eu tentava sempre ser ativa na parte de design, ajudando na pesquisa e parti-

cipando dos testes de usuários. Isso fazia com que eu sempre estudasse design mesmo estando focada em código durante o primeiro ano. Além disso, algo que me ajudou muito foi desenvolver projetos pessoais e como nesses projetos eu sempre desenvolvia desde a estratégia até a implementação eu me mantinha aprendendo tanto design quanto código.

Levy: Durante o Academy eu trabalhei em projetos paralelos em que eu concebia desde o design ao desenvolvimento. Quando eu desejava aprender algo criava para mim mesmo um desafio e só parava de estudar quando eu concluísse este desafio. 📌

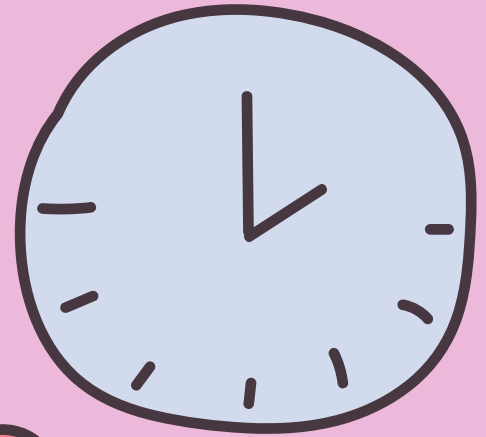
“EU ACREDITO QUE A GENTE PRECISA FOCAR NO PRESENTE, O QUE VOCÊ QUER HOJE?”

HISTÓRIAS {

Tic Toc

Trabalho e família

TEXTO *Patrícia*
EDIÇÃO *Kellyane*
DESIGN *Carolina*
ILUSTRAÇÃO *Carolina*



Tic Toc

TIC

TOC

BLA

A dedicação ao trabalho é importante para o sucesso profissional e uma ótima maneira de ser exemplo para sua família. Mas é preciso cautela nesse aspecto, pois devido a ansiedade de ser destaque em sua carreira profissional ou mesmo achando que este fato lhe tornará feliz, você acaba negligenciando aqueles que ama. A pandemia trouxe o ambiente de trabalho para dentro de casa e não ter mais a divisão desses dois ambientes pode parecer confuso. Portanto vamos discutir soluções e trazer essa problemática a nosso favor nesse artigo.

Uma das dificuldades mais frequentes entre pessoas que não equilibram o trabalho e a vida familiar são as horas extras de trabalho. Por isso, tente aproveitar da forma mais eficiente possível sua jornada de trabalho, mantendo toda sua concentração nas tarefas profissionais. Fazendo de cada minuto o mais produtivo possível, então as chances de você concluir suas atividades no horário estipulado e estar com sua família nos horários esperados são maiores. Evite os pequenos “ladrões do tempo”, como reuniões mal preparadas, distrações na internet, excesso de conversas improdutivas, etc. A técnica do pomodoro pode ser uma maneira fabulosa de ajudar nesse aspecto. A técnica se baseia na ideia de que dividindo o nosso fluxo de trabalho em blocos de concentração

intensa, conseguimos melhorar a agilidade do cérebro e estimular nosso foco. Vale a pena pesquisar sobre esse assunto, fica a dica!

Estabelecer prioridades sempre tornará suas atividades mais produtivas. Já que sua família é mais importante do que seu trabalho, porque não utilizar dos meios de agendamento de compromissos importantes com os eventos familiares também? Aprenda a valorizar esses momentos e diga “não” a convites ou eventos de trabalho que sejam dispensáveis para sua carreira profissional ou que coloquem em risco momentos especiais da sua vida familiar. Você sempre poderá agradecer o convite, alegando que já tem um compromisso.

O ideal é desconectar-se totalmente quando estiver fora do horário de trabalho. Isso significa, por exemplo, não verificar o e-mail ou grupo de mensagens do trabalho para valorizar o tempo e dedicar sua atenção por completo aos seus familiares.

Não é justo estar mal humorado por causa do trabalho e não compartilhar nada. Converse com seu companheiro sobre o que está se passando. Seu parceiro é a pessoa mais adequada a lhe escutar e aconselhar, já que deve ser quem melhor lhe conhece e quem você deve confiar. Falar sobre o trabalho pode evitar muitos conflitos. Se a pessoa que passa todos os dias ao seu lado



sabe pouco ou nada sobre o que ocorre durante o seu horário de trabalho, ele provavelmente ficará desorientado quando você, eventualmente, se comportar de modo mais agressivo ou retraído sem explicações.

Aproveite o tempo em casa para realmente viver em família. Muitas pessoas mergulham na televisão, na internet, nos jogos ou celular, deixando de dedicar-se aos familiares. A hora das refeições, por exemplo, deve ser um momento de encontro, em que todos os membros da família sentam-se à mesa para compartilhar as experiências. São momentos de qualidade vividos que levamos conosco para sempre em forma de laços afetivos.

Como você percebeu, conciliar trabalho e família não é inalcançável, basta ter organização. Se você souber equilibrar a sua vida profissional, logo sobrar tempo para dedicar à sua vida pessoal. Seguindo alguns desses passos, você conseguirá se dedicar para a sua família sem prejudicar o seu trabalho. ❶

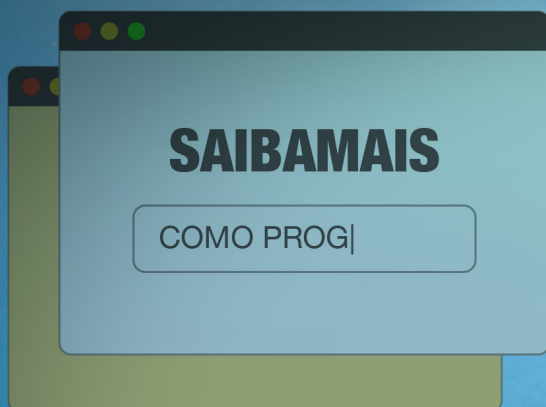
PROCESSOS {

Mergulhando na Web

TEXTO Kellyane

EDIÇÃO Tales

DESIGN Carolina



Não é incomum nos perguntamos sobre algo de difícil explicação ou surgir aquela dúvida repentina sobre um assunto qualquer. Quando isso acontece, a internet pode ser nossa grande aliada. A verdade é que estamos em um período benéfico para aqueles que querem uma resposta rápida, pois com apenas alguns cliques podemos encontrar uma infinidade de coisas.

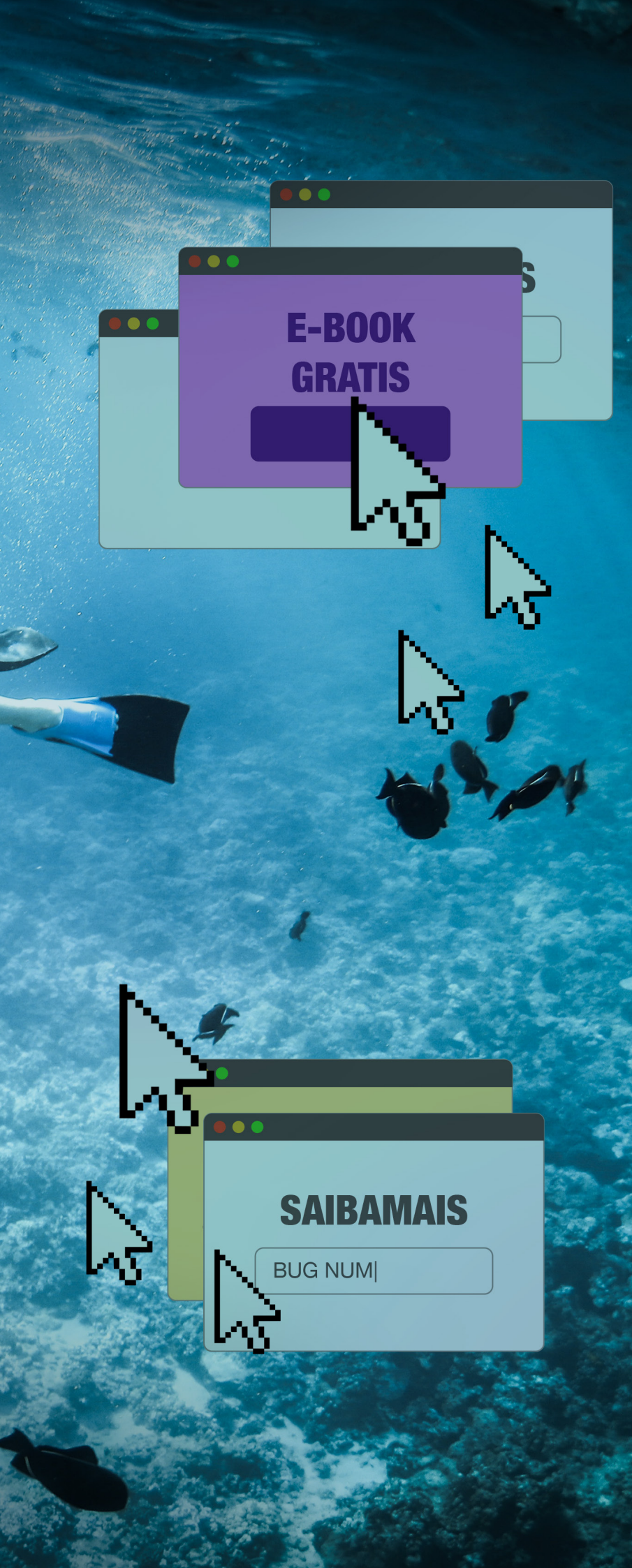
Mas, assim como é fácil nos depararmos o tempo todo com as pesquisas, também é frequente nos perdermos diante de tanto conteúdo. Desta forma, faz-se necessário adotarmos bons hábitos que possam organizar este conteúdo e deixá-lo a nosso gosto.

As ferramentas de busca que conhecemos como, por exemplo, Google e Yahoo, utilizam o que chamamos de palavras-chave. Significa que cada palavra digitada na busca será como um índice para que o algoritmo procure as páginas que as contém ou tem alguma relação com elas. Tendo isto em vista, a primeira coisa a se fazer é pensar claramente sobre o que você procura e tentar resumi-lo em poucas palavras. Eu sei, organizar as ideias também pode ser um trabalho complexo dependendo da pesquisa, então vamos as coisas que podem facilitar nossa vida!

Filtrar a busca

Há várias maneiras de filtrar uma pesquisa e nós começaremos com o mais simples. Tentaremos nos basear na ferramenta Google, já que é a mais utilizada. Você já reparou que próximo a barra de pesquisa do seu navegador a botões intitulados com “Imagens”, “Vídeos” e etc? É um bom começo selecionar o tipo de sua pesquisa, acredite, pode eliminar muitas páginas web irrelevantes para você. Em casos de pesquisa por imagens, é só colar a url desta imagem na barra de pesquisa e logo encontrará imagens semelhantes a que você tem, que se encaixam no seu contexto. E em ferramentas de pesquisa, você consegue filtrar ainda mais podendo definir tamanho, cor e uma série de outras coisas.

Agora que temos nossa área de pesquisa definida, vamos escrever exatamente o que queremos. Mas como? Bom, se você deseja páginas com um trecho ou frase específica utilize as aspas na sua busca, é muito útil quando se quer procurar por textos de livros ou letras de músicas. Você também pode usá-las para pesquisar algo dentro de um site específico, que já é mais uma dica de tornar sua busca mais eficiente. Basta colocar a palavra site: e inserir o site desejado depois dos dois pontos. Aí vai um exemplo: **site:developer.apple.com “design”**.



Outra coisa, caso deseje pesquisar um assunto em um determinado tipo de arquivo, você também consegue fazer isso facilmente usando filetype: como por exemplo, Inteligência Artificial filetype: .pdf. Ao invés de acrescentar, você também pode retirar termos com o sinal de subtração, é muito útil quando há o mesmo termo em mais de uma área, por exemplo, ASO -saúde.

Organizar o conteúdo

Com nossas páginas filtradas, podemos agora organizar todo nosso conteúdo de acordo com o que nos é mais propício. Uma das formas de organizar é favoritar quais páginas são mais úteis para você no próprio navegador para que possa retornar a elas quando necessário. No Safari, basta pressionar o + que fica ao lado da Url e no Chrome, à direita da barra de endereço, clique na estrela. Você também pode optar por aplicativos que possam salvar e categorizar suas páginas como Pocket e Flipboard ou até mesmo possa anotar observações e acessá-las de qualquer dispositivo como o Aplicativo de notas da Apple ou Google Keep.

E assim, fica bem mais prático encontrar o que se deseja não é mesmo?! Agora estamos prontos para mergulhar na web sem se afogar. 📌

CÓDIGO {

Por que eu devo me preocupar com a LGPD?

TEXTO *Alley*
EDIÇÃO *Tales*
DESIGN *Carolina*

A Lei Geral de Proteção de Dados (LGPD) entrou em vigor em setembro de 2020, prometendo ser a carta magna para proteger as informações pessoais e sensíveis dos internautas brasileiros. Somos bombardeados diariamente com notícias a respeito de novos vazamentos de dados e violações de privacidade; nem sempre esses vazamentos são ocasionados por hackers, na maioria das vezes, tudo o que basta é o próprio administrador de sistemas cometer um erro e esquecer um backup de todo o banco de dados de seu sistema em um diretório aberto indexado pelo Google, ou cometer qualquer outro erro simples como por exemplo, expor dados de usuários em planilhas abertas com senhas, CPF e outros dados no GitHub como recentemente ocorreu no Ministério da Saúde. Parece absurdo demais? Pois é o que acontece o tempo todo. Uma pesquisa no google pode trazer dados relevantes sobre a amante do vizinho, seu colega de trabalho, você mesmo, etc. Sites como o Reclame Aqui são inundados por queixas de clientes a respeito de supostas infrações a privacidade de dados (na maioria das vezes, contatos insistentes de equipes comerciais que não recolheram o consentimento dos internautas). Sabe aquela ligação que você recebe todo dia de algum telemarketing? Exatamente. Isso ocorre porque seus dados foram vazados ou vendidos para alguma empresa de telemarketing sem você saber. Felizmente,

com a LGPD toda essa situação deve melhorar.

Má configuração de servidores e de ambientes web são problemas cada vez mais comuns, especialmente em épocas de transformação digital acelerada por conta da pandemia do COVID-19. Sabe o UserDefault usado para salvar dados sigilosos como senha? Hmm, temos um problema. É possível que um iPhone com jailbreak acesse facilmente qualquer dado que foi salvo lá.

A Apple usa o ATS justamente para amenizar os riscos de conexões que não atendem às especificações mínimas de segurança, porém se você passar um nome de usuário e senha sem criptografia em uma requisição POST, mesmo com o ATS habilitado, um simples proxy consegue visualizar o conteúdo da requisição, afinal, apesar de amenizar muitos problemas, o ATS não faz milagre.

NEM TODO VAZAMENTO DE DADOS É OCASIONADOS POR HACKERS, NA MAIORIA DAS VEZES É CULPA DE QUEM ADMINISTRA O SISTEMA.

foto: Reimund Bertrams

E ENTÃO, COMO ENTRAR EM CONFORMIDADE COM A LGPD?

I. SISTEMA SEGURO

A chegada da LGPD não promoveu uma lista de frameworks certos ou errados. O que realmente importa é se tudo o que foi definido no projeto foi planejado, documentado e implementado pensando na privacidade dos dados do usuário. Obviamente isso não significa que em nenhum momento do seu trabalho você não poderá alterar as ferramentas ou mesmo as infraestruturas, mas será preciso garantir que estas sejam reconhecidas e documentadas.

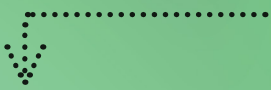
2. CONSENTIMENTO DO USUÁRIO E REQUISITOS DE PROJETO

Rever a proteção de dados, levando em conta as diretrizes da lei, significa passar o controle dos dados aos usuários utilizando de mecanismos de consentimento e acesso de sujeito. A partir daí o usuário deve ser informado a respeito do fluxo de seus dados e de seus direitos sobre os mesmos. Minha sugestão é que você nunca solicite dados do usuário a não ser que seja estritamente necessário. Mas caso o faça, você precisa deixar claro do motivo dessa solicitação, e precisa garantir a confidencialidade dos dados coletados. É necessário também garantir que o usuário possa alterar e excluir informações.

3. DOCUMENTAÇÃO E TESTES SÃO A CHAVE

A regra de ouro da LGPD é documentação e testes. Você precisará comprovar em situações extremas de vazamento que você fez ambos, documentação e testes, da melhor forma possível no período em que o desenvolvimento ou manutenção do sistema aconteceu. O que não estiver registrado, então é porque não aconteceu.

Por fim, lembre-se: a privacidade dos dados necessita ser protegida a cada passo do usuário, do login ao logout. Uma documentação bem feita, pode te salvar de um futuro processo. ⓘ



VOCÊ PODE PEDIR DADOS DO USUÁRIO, DESDE QUE A FINALIDADE DA AQUISIÇÃO DELES SEJA APRESENTADA DE FORMA CLARA!

HISTÓRIAS {

Mulheres na tecnologia

TEXTO *Patrícia*
EDIÇÃO *Beatriz*
DESIGN *Carolina*
ILUSTRAÇÃO *Carolina*



Existem muitos tabus sobre as mulheres na área da tecnologia que precisam ser discutidos e desmitificados. Um desses reflete a temática desse artigo, então eu questiono a você meu leitor: Porque não existe um nome verdadeiramente popular feminino na tecnologia? Sabemos que existiram muitas mulheres incríveis na história e atualmente vemos trabalhos fantásticos feitos por elas, então porque dificilmente os nomes dessas mulheres não aparecem na mídia?

E a resposta para essas e outras questões pode estar em como a nossa sociedade pensa e define o que seja masculino e feminino durante o desenvolvimento das habilidades e competências desde a infância. A escola já naturaliza a concepção dos estereótipos, delegando às garotas tarefas de cuidado e sociais. Consequentemente, expectativas são mais baixas para as meninas em relação ao aprendizado das disciplinas de exatas, por exemplo, ocasionando medo e ansiedade, portanto bloqueando o aprendizado. Existe muito a crença do “Quem é bom no que...” e apesar de serem tão capazes quanto os garotos, as meninas acabam perdendo a confiança no aprendizado e não se destacando em matérias como a matemática. A confiança do aluno em aprender influencia de fato em seu engajamento na matéria e quando o mesmo sai

da escola, vemos o quanto essa habilidade é importante para enfrentar os desafios do mercado de trabalho.

No ambiente acadêmico não é muito diferente. Notamos um ambiente pouco acolhedor para as mulheres, portanto refletindo em uma elevada taxa de evasão e desinteresse por parte das mesmas. Percebemos casos de alunas que são simplesmente ignoradas ou subjugadas por professores, somente pelo fato de serem mulheres com o pensamento machista delas não terem capacidade de estar ali. Do mesmo modo, mulheres negras são duas vezes mais afetadas, pois infelizmente o mundo da tecnologia se mostra efetivamente masculino e branco.

Os desafios para as meninas que seguem na carreira são muitos e dentre eles, os salários menores. As mulheres ganham 30% menos do que homens¹. E, uma das análises aponta que as mulheres ganham um salário inferior, visto que, são menos propensas a negociar seus salários depois de uma oferta inicial.

Negociar salários geralmente é encarado como conflito, no entanto, para os homens é um processo natural.

Portanto, precisamos compreender toda a dinâmica histórica da construção das profissões na área da tecnologia para modificar pensamentos, desconstruir estereótipos e ganhar espaços, mostrando talentos e valorizando-os cada dia mais para enfrentar esse desequilíbrio atual e modificar o futuro da área da tecnologia. Nesse cenário, inclusão e participação feminina dos espaços trarão novos olhares e percepções importantes para a produção de tecnologia. E nesse sentido, precisamos que as empresas construam espaços de trabalho que rejeitem o machismo, o sexismo, a misoginia e estimulem a igualdade de gênero.

**TABUS SOBRE AS MULHERES NA
ÁREA DA TECNOLOGIA PRECISAM
URGENTEMENTE SER DISCUTIDOS
E DESMISTIFICADOS.**





Katie Bouman

Professora assistente de ciência da computação no Instituto de Tecnologia da Califórnia. Ela foi responsável no MIT (Instituto de Tecnologia de Massachusetts) por um algoritmo usado na criação das primeiras imagens de um buraco negro, publicado em abril de 2019, fornecendo suporte computacional para aprender sobre a relatividade geral no regime de campo forte.



Carol Shaw

Primeira mulher a brilhar na indústria dos games, Carol Shaw foi uma das funcionárias originais da Atari. Porém foi na Activision, participando do desenvolvimento de um dos maiores clássicos dos games, River Raid que foi reconhecida. Estes são alguns games clássicos que ela trabalhou: 3D Tic Tac Toe, Super Breakout e Happy Trails.



Jean Sammet

Criadora do FORMAC que é uma das primeiras linguagens de programação, ela entrou em uso no final dos anos 1960 pelas mãos da IBM. Era utilizada para trabalhar fórmulas matemáticas e auxiliar em cálculos complexos. Sammet também influenciou na criação do COBOL e participou de muitas entidades de inclusão das mulheres da tecnologia.



Roberta Williams

Cofundadora da On-Line Systems, que depois, se tornaria a Sierra, um dos mais importantes nomes da indústria de jogos eletrônicos, tendo participado diretamente do desenvolvimento e distribuição de grandes jogos como King's Quest, Phantasmagoria, Half-Life e Counter-Strike.



“As garotas do ENIAC”

Durante a Segunda Guerra Mundial, Betty Snyder, Marlyn Wescoff, Fran Bilas, Kay McNulty, Ruth Lichterman e Adele Goldstine foram responsáveis pela configuração do ENIAC. Elas lidavam diariamente com mais de três mil interruptores e botões do hardware de 80 toneladas. Elas foram responsáveis por dar o criar muitos protocolos usados até hoje.



Ada Lovelace

Foi matemática e escritora inglesa. Conhecida por ter escrito o primeiro algoritmo para ser processado em uma máquina, chamada a máquina analítica de Charles Babbage. Esses algoritmos permitiam a máquina computar os valores de funções matemáticas. Ada já se questionava como as pessoas e a sociedade se relacionam com a tecnologia como uma ferramenta de colaboração.



Radia Perlman

Radia foi responsável pela criação do protocolo STP. Esse protocolo de rede otimizou os sistemas conectados evitando os loops de dados. Dessa forma garantiu que as informações trafegassem mesmo em caso de falhas. Ela também é uma das pioneiras no ensino de programação e arquiteturas de redes para crianças. Hoje trabalha na Intel.



Frances Allen

A primeira mulher a ganhar o Turing Award, Allen trabalhou durante 45 anos na IBM, onde focou seus trabalhos em levar as máquinas para as casas de pessoas comuns. Além disso, ajudou a criar alguns dos primeiros sistemas de segurança da agência de segurança nacional do governo dos EUA. Muitos de seus trabalhos no setor de inteligência não foram divulgados por questões de sigilo. ❶

A importância dos testes de usuário

TEXTO *Ronaldo*

EDIÇÃO *Beatriz*

DESIGN *Carolina*

ILUSTRAÇÃO *Felipe*



Este não é um texto de design para designer, longe disso. É escrito por alguém que se interessa por programação, porém percebeu que testes de usuário influenciam no desenvolvimento do código. Através da experiência de um projeto real, mostrarei como testes de usuário foram importantes no desenvolvimento do projeto. A ideia era a criação de um aplicativo sobre astronomia, com leituras e imagens sobre corpos celestes. Nesse contexto, os testes de usuário eram um requisito obrigatório, e no início tinha pouca experiência de como executá-lo. No entanto, mal sabia que iria mudar o rumo da aplicação e o quanto iria me impressionar, pois evidenciou pensamentos e ideias que somente o usuário poderia identificar.

Primeira rodada de testes

No início do projeto, não tinha planejado em que momento iria executar os testes, porém resolvi seguir o iceberg de UX e quando chegou no wireframe decidi fazer os primeiros testes. Após a realização dos testes, eles me possibilitaram enxergar que a usabilidade do aplicativo não estava acessível como imaginei ou agradáveis para o usuário. Percebi que cores e fontes foi o que mais mudou no projeto a partir das opiniões dos testes. Nesse momento já fiquei animado com os resultados e o

que eles me mostraram no protótipo de alta fidelidade. Analisando todo o processo pude perceber que se existe um momento ideal para realizá-los é entre as etapas de wireframe e de alta fidelidade. Dentre os motivos, o mais forte, na minha visão como desenvolvedor, é que como ainda não tem código é mais fácil executar alterações no escopo geral, o que facilita o processo de desenvolvimento do aplicativo como um todo, logo, testar nesse momento é fundamental.

Segunda Rodada de testes

O segundo momento que realizei testes de usuário foi no final do projeto com o aplicativo desenvolvido e numa versão estável. Nesse momento, o objetivo era testar o funcionamento do aplicativo, o comportamento do usuário, a usabilidade e acessibilidade. Na época não sabia, mas com estudos posteriores sobre o assunto vejo que realizei testes de usabilidade nessa segunda rodada, pois ele consistiu em deixar o usuário usar o aplicativo de forma remota, comigo assistindo o uso dele, mas sem interferir deixando-o livre. Após 2 a 3 minutos, o usuário me passava um feedback geral. Esses testes foram incríveis, pude perceber uma série de otimizações que estavam “na minha cara” e eu não enxergava mas que o usuário usava-

do o aplicativo por 3 minutos conseguia ver. Melhorias como elementos que eram clicáveis, porém não eram intuitivos o bastante para o usuário clicar ou imagens que poderiam ser arrastadas para o lado e mostrar novas imagens que também não davam essa impressão, além de uma série de dicas de melhorias em geral. Portanto, esse segundo teste foi mais revelador que o primeiro por ter essa visão que o usuário tinha e eu como criador do aplicativo não.

Para finalizar

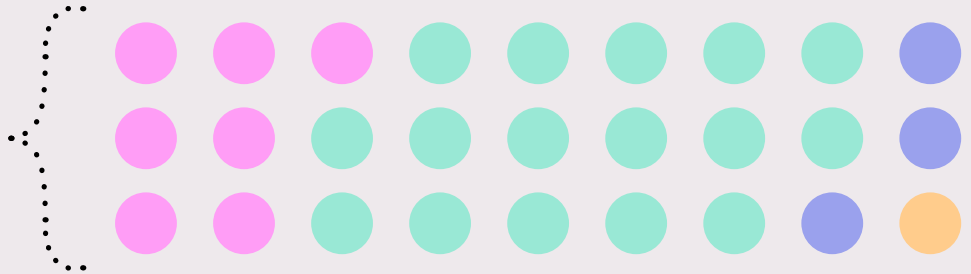
Os testes de usuário são uma tarefa essencial em um projeto e não precisam ser necessariamente papel do designer. Desenvolvedores podem e devem participar desse processo, opinando e principalmente se mostrando abertos a feedbacks do usuário, dado que essas pessoas são o público alvo do aplicativo. Para o desenvolvedor uma ótima analogia sobre esta importância é relacionar com testes unitários. No código fonte de uma aplicação apenas conseguimos garantir que um método funcione perfeitamente depois de fazer testes unitários de qualidade. Da mesma forma, os testes de usuário consegue garantir que a interface do seu aplicativo está com usabilidade!



INFOGRÁFICO {

2020 em números

135



SOLUTIONS

35 solutions
não-apps

80 protótipos
funcionais

15 apps em
testflight

5 apps
na loja

19

FORMULÁRIOS DA HORA DO ALMOÇO

198

DIAS LETIVOS

36

WORKSHOPS APPLE+MENTORES

28 dias
presenciais

170 dias
à distância

51

LEARNERS

Incluindo todos os
40 alunos da turma
regular, 7 mentores
e 4 pessoas do corpo
tec. e administrativo

423MIL

MINUTOS NO ZOOM

+6MIL

COMMITTS NO GIT



