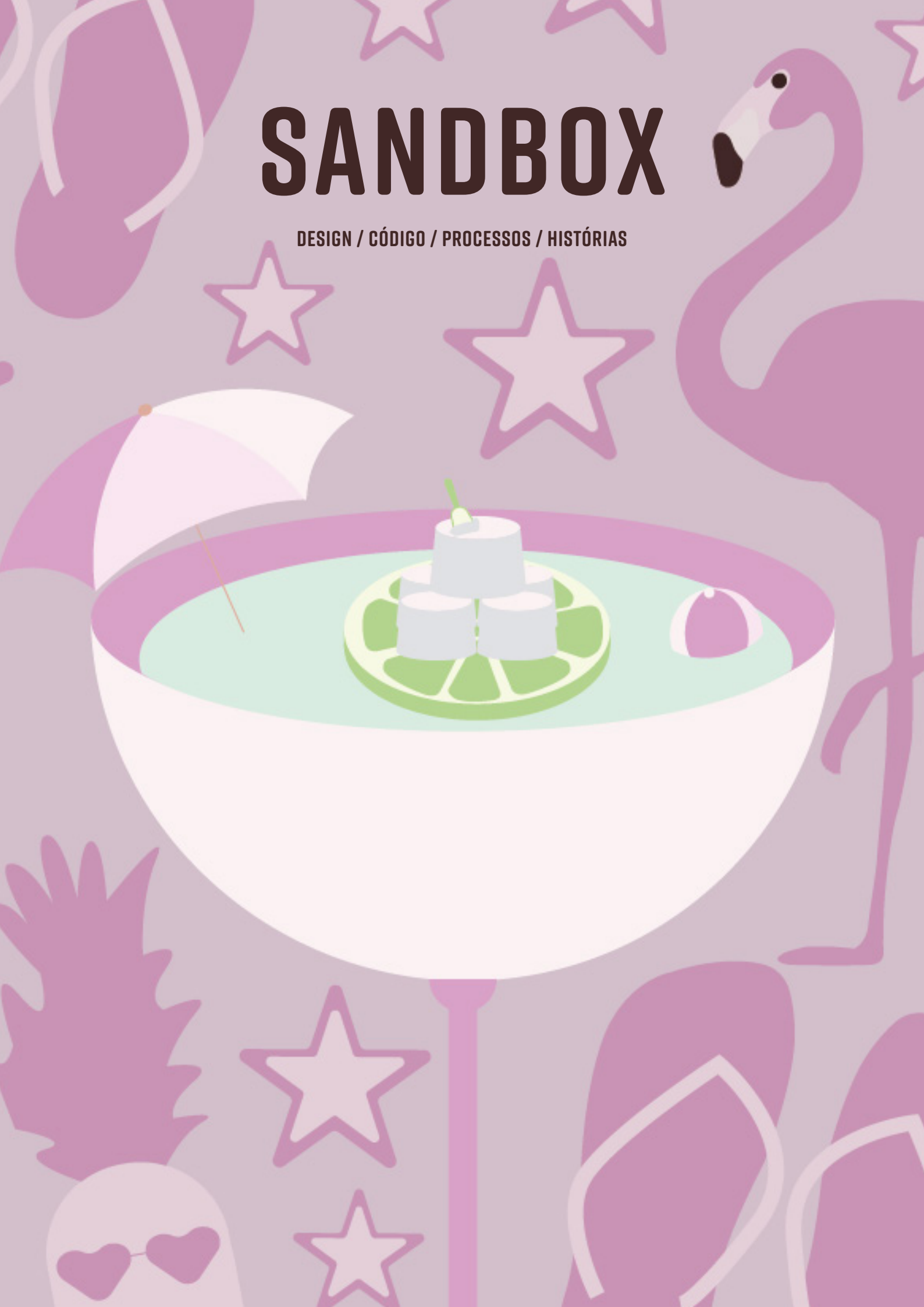


SANDBOX

DESIGN / CÓDIGO / PROCESSOS / HISTÓRIAS



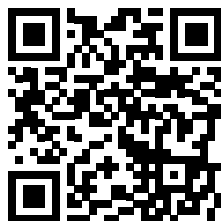
ACADEMY IFCE

Equipe de redação: Alley Pereira, Ana Karolína da Silva, Carolina Barbalho, Cecília Soares, Elias Ferreira, Elis Vieira, José João Silva, Kellyane Nogueira, Leonardo Gomes, Mateus Nobre, Nathália Moura, Patrícia Sampaio, Ronaldo Gomes, Tales Conrado, Livia Oliveira, Ravena Amaral, Cristiano Araújo e Higor Oliveira **Equipe de edição:** Carolina Barbalho, Beatriz Carlos, Brenna Amorim, Helaine Pontes, José Mateus, Kellyane Nogueira, Mateus Rodrigues, **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, Elis Vieira e Carolina Barbalho **Projeto gráfico:** Carolina Barbalho

Revista “Sandbox”. Edição 002. De Junho de 2021. “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. Todos os produtos apresentados nas ‘propagandas’ desta revista foram desenvolvidos pelos alunos do Academy IFCE. **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 **12.5** e **5.4** respectivamente. Consultar o histórico de atualizações caso esteja trabalho com uma versão diferente destas.

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



Algumas pegadas não se apagam

Durante essas semanas eu me deparei com um bloqueio criativo absurdo para escrever esse editorial. Afinal, o que diabos de tão interessante eu poderia escrever para ser o primeiro conteúdo a ser lido na revista?

O nome da revista bateu e rebateu na minha mente diversas e diversas vezes “sandbox, sandbox.... SANDBOX!”. E foi de tanto pensar na caixa de areia, que me veio a memória perfeita para esse editorial:

Quando eu tinha os meus 7 ou 8 anos, gostava muito de ir a praia e fazer castelos de areia, enterrar meu irmão na praia, soltar pipa, correr feito louco... mas acho que uma das coisas mais legais que eu gostava de fazer era cavar buracos enormes e ficar dentro feito piscininhas. Nessa época, eu definitivamente não sabia que existiam várias praias no mundo. E quando avistava alguma piscininha eu pensava: “olha ai, eu que fiz e ainda tá ai!”. Esse pensamento se internalizou em mim em formato de sentimento e hoje frequentemente eu penso a respeito disso. Não sobre castelos de areia ou piscininhas na praia e sim sobre legado.

Legado é algo que pode ser

transmitido de gerações em gerações, como um valor ou algo do tipo. E com isso, caro leitor, eu pergunto: você já parou para pensar qual o legado que você está deixando por onde você passa? Ou qual o legado o lugar por onde você passou deixou em você?

Quero deixar claro, que não acredito que isso seja algo que deva ser cobrado de todos, afinal cada um trilha a sua jornada da sua maneira. Mas vamos refletir um pouco a respeito disso.

Acredito que é bem claro qual o legado que os Academies do Brasil conseguem deixar na vida dos seus estudantes:

Desenvolvimento. Tanto das *hard skills* como nas *soft skills*.

Mas o que sempre penso é: como eu posso deixar o meu legado aqui? Como as próximas gerações vão saber que eu estive aqui um dia? Como posso deixar minha marca?

E esse pensamento me perseguiu muito no dia que decidimos “Hey gente, vamos fazer a segunda edição da Sandbox?”. E acho que é por isso que esta edição tem um sabor tão especial para mim. Quando comecei a pensar no quão atemporal são os conteúdos que as pessoas produziram para essa edição,

o sentimento de quando eu era criança veio bem forte.

Pois não importa se daqui a 10 anos alguém que leia esta revista perceba que alguma tecnologia apresentada está obsoleta.

Ela é atemporal pois o conteúdo aqui apresentado consegue ser passado de geração em geração, consegue mostrar como a tecnologia era usada e o quanto ela evoluiu e também consegue eternizar os aprendizados da nossa turma.

Porque todos que participaram desta edição conseguiram deixar o seu legado ou sua marca de alguma forma: contando suas próprias experiências de desenvolvimento, compartilhando conhecimentos técnicos sobre código, design, processos ou sugerindo dicas.

Nesse momento, ainda em pandemia, a Sandbox vem como um símbolo de resistência de que não paramos, de que não conseguimos parar, de que não vamos parar nunca. Vem com um sentimento muito forte de inquietação, sempre em busca de fazer melhor que antes: queremos mostrar ao mundo tudo que já fizemos e do que somos capazes. Esse de fato é a nossa marca.

O nosso legado. ❶

José João Silva
COORD. ILUSTRAÇÃO / EMBAIXADOR

CONTEÚDO {



06

↳ O MANUAL COMPLETO PARA ASSUSTAR SEUS USUÁRIOS

Para as que vezes em que maior *user need* é levar um sustinho...

08 LETRANDO ABC

Um aplicativo de alfabetização com realidade aumentada!

14 LETRANDO ABC

Um aplicativo de alfabetização com realidade aumentada!



18

↳ DESMISTIFICANDO O TIME DE DESENVOLVIMENTO SCRUM

Vamos explicar a sopa de letrinhas do Scrum. PO, PM, SM...

20 GRAND CENTRAL DISPATCH

Aprenda a gerenciar tarefas no código executadas em paralelo.

26 CSI: CASE STRUCTURING INVESTIGATION

O trabalho de um designer muitas vezes é digno de um verdadeiro investigador...



28

↳ LISS: UM ESPAÇO SEGURO PARA A APRENDIZAGEM E INOVAÇÃO

Um projeto de educação que cultiva um novo modelo de aprendizagem para a educação!

36 TESTES UNITÁRIO X SINGLETON

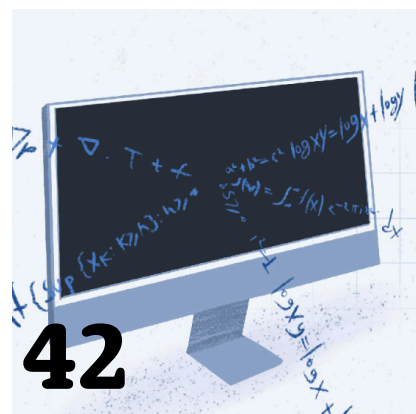
há quem ame e quem odeie, mas todos sabem que é importante.



38

↳ CRIATIVIDADE: QUE TIPO DE HABILIDADE É ESSA?

De onde ela vem? Para onde ela vai quando eu preciso? Vamos entender hoje no... Sandbox!



42

↳ A IMPORTÂNCIA DA MATEMÁTICA PARA DESENVOLVEDORES

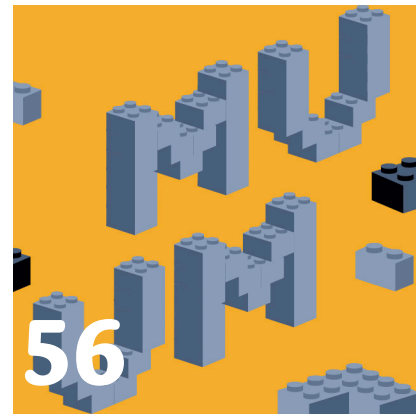
Para as que vezes em que maior *user need* é levar um sustinho...

46 UM MAPA DA UX RESEARCH

Para te ajudar a encontrar seus usuários e o melhor produto!

50 SONDA CULTURAL

Pesquisa co-criativa, qualitativa e semi-etnográfica em Recife!



56

↳ DESIGN PATTERNS: O QUE SÃO E QUAIS OS BENEFÍCIOS

Mais sobre este tema tão importante para o design de software!

60 QUEM É O + QUERIDINHO?

A guerra de opinião entre Xibs, Storyboard e View Code!



Vacinaí

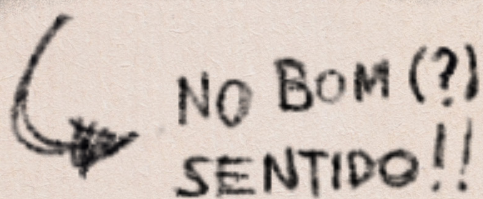
Nunca mais perca seu cartão de
vacinação, pois agora ele pode
ser digital!

Nos siga no Facebook e para conferir as
novidades e apoiar nossa ideia! Este é um
aplicativo de saúde sem fins lucrativos



DESIGN {

O manual completo para assustar seus usuários



TEXTO *Felipe*
EDIÇÃO *Carolina*
DESIGN E ILUSTRAÇÃO *Carolina*

Imagine você com seus 12 anos. Após um dia de estudo e muita traquinagem, você toma seu banho e se prepara para ir para cama. Seus pais já estão dormindo, mas bate aquela sede. Com a casa toda escura, você caminha sozinho em direção a cozinha, acende a lâmpada do corredor, acende a lâmpada da cozinha, bebe sua água e devolve o copo à pia... E então o grande o momento chega: você precisa apagar as lâmpadas e voltar para o seu quarto.

Você apaga a luz da cozinha, e olha para trás... está tudo bem. Você anda pelo corredor meio apreensivo, chega no interruptor, dá mais uma olhada para o corredor por onde você veio e com o olhar fixo, apaga a luz. Ufa! Acabou. Uma última olhada na sala e, o que é aquilo contra a parede? Uma sombra, uma silhueta que não estava lá antes. Seu coração acelera, um frio toma conta da sua barriga, sua pupila se dilata, você fica ofegante e precisa decidir se vai correr pra cama ou acender a luz de novo e tentar entender o que está acontecendo. É uma descarga enorme de substâncias neuroquímicas e hormônios.

Esse é um caso típico de medo. O estado afetivo suscitado pela consciência do perigo. É uma emoção que herdamos dos nossos antepassados e que foi necessária para sobrevivência da nossa espécie. Essa emoção serve como um alarme de defesa que prepara o corpo para reagir às possíveis situações que ameacem nossa existência. Essa ameaça pode ser real,



ou não. Mas isso não importa. O nosso corpo vai estar preparado.

Esse estado de alerta serve para obtermos o máximo de informações para que possamos reagir da forma mais rápida possível a qualquer ameaça, é uma característica evolutiva da nossa espécie para aumentar a taxa de sobrevivência em situações de risco.

E, apesar de parecer estranho, muitos buscam ativamente sentir a emoção de medo como entretenimento. Talvez você mesmo seja uma destas pessoas. O fato é que na maioria das vezes quem busca este tipo de experiência busca algo muito semelhante aqueles que praticam esportes: experienciar as altas doses de substâncias como adrenalina.

Mas como tudo, o medo também deve ser sentido com moderação.

O ritmo do terror

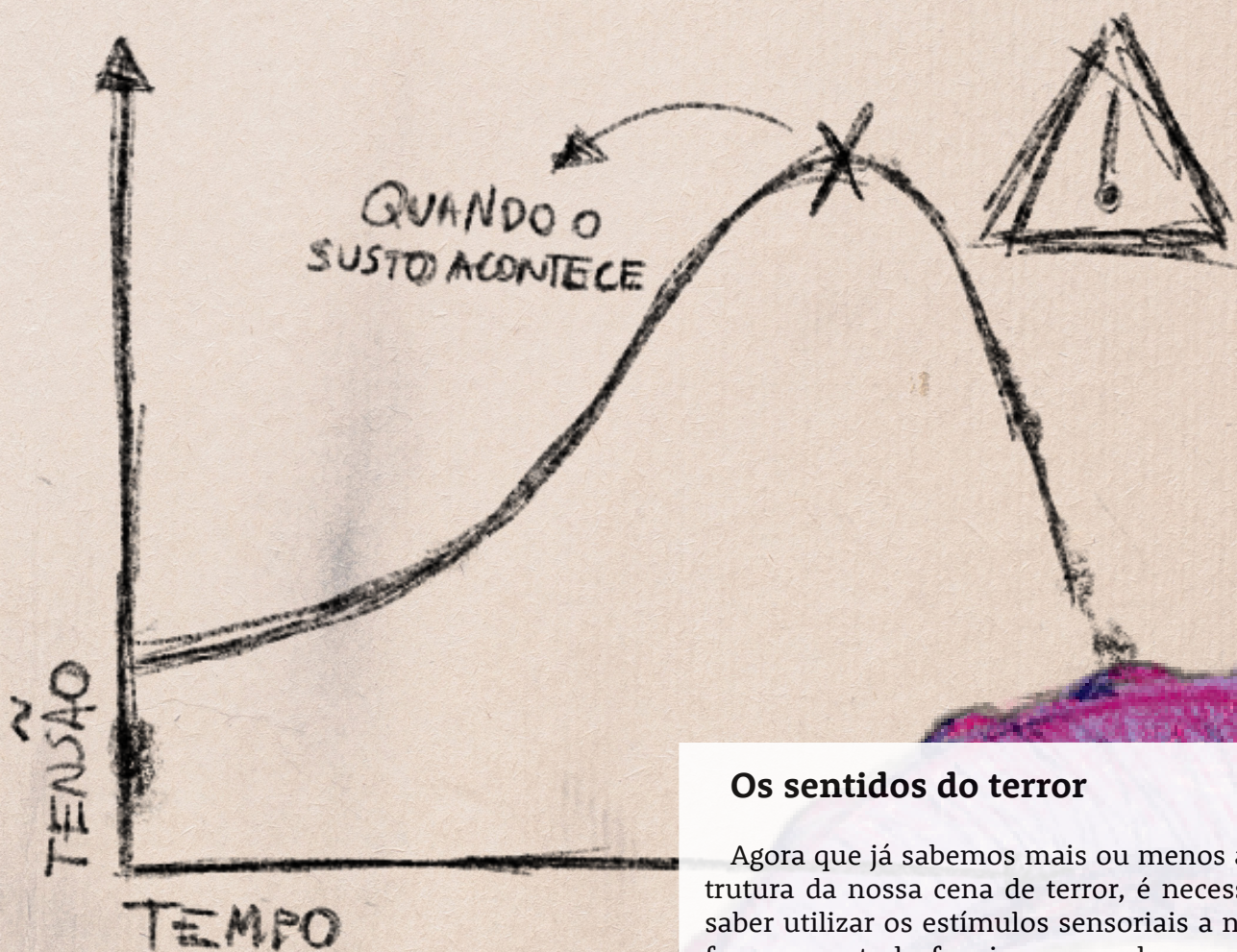
O terror se baseia em criar tensão até um pico quase insuportável, e então... BUM! Algo acontece! O monstro aparece de repente, um gato pula do arbusto, as luzes se apagam. E logo em seguida temos um relaxamento. Pois não importa o quão assustador seja o monstro que está agora na tela, nada vai superar os momentos de suspense logo antes da sua aparição.

E este ciclo se repete durante toda a experiência. Sempre com essa tensão sendo construída de forma lenta até chegar em um pico, com uma ação importante e logo em seguida, temos um momento de relaxamento.

E o que cria a tensão da cena é a utilização do desconhecido e do estranho. Tudo que não temos conhecimento ou controle nos ameaça em certo nível, pois não sabemos se aquilo pode ou não nos matar. Já o estranho se refere a qual-

quer coisa que não funcione da forma esperada.

Um exemplo clássico de acontecimento que explora a estranheza é uma luz falhando. Esse não é o comportamento esperado dela e isso significa que algo não está certo. Não tendo conhecimento do que pode estar causando esse mal funcionamento, podemos ficar com medo (ainda mais quando estamos vendo um filme de terror onde qualquer coisa pode ser motivo de preocupação).



Os sentidos do terror

Agora que já sabemos mais ou menos a estrutura da nossa cena de terror, é necessário saber utilizar os estímulos sensoriais a nosso favor, para tudo funcionar e podermos criar verdadeiras cenas de tensão.

E, dependendo da mídia que você irá utilizar, diferentes sentidos poderão ser explorados. Nos filmes, por exemplo, nós só podemos explorar dois sentidos: visão e audição. Já no caso de jogos em consoles ou mobile, também temos o tato trabalhando a nosso favor. A partir das vibrações, ou feedbacks hápticos.



A AUDIÇÃO PODE SER EXPLORADA, MAIS QUE A VISÃO, PARA CRIAR A SENSAÇÃO DE INSEGURANÇA E ESTRANHEZA

Audição

A audição é a grande responsável por criar uma atmosfera carregada emocionalmente. Ela pode ser explorada, mais que a visão, para criar a sensação de insegurança e estranheza ao escutar sons como passos em uma casa vazia, objetos caindo e outros que podem ser ótimos para iniciar a construção da tensão. De acordo com o trabalho de Daniel Blumstein na Universidade da Califórnia (UCLA), existem semelhanças entre a gama de notas altas e baixas em filmes de terror e os sons que os animais produzem em situações em que estão expostos ao medo ou ao ataque de predadores. Para falar sobre sons em peças de terror, primeiro precisamos entender que existem mais de uma categoria de sons, estas são: trilha sonora, efeitos especiais, melodia e ruído. Cada um desses tipos de áudio tem suas funções e devem ser aplicados de acordo com o contexto da cena. Não faz sentido, por exemplo, em uma cena onde a tensão está começando a ser construída utilizar uma trilha sonora agitada com diversos instrumentos tocando juntos em um ritmo acelerado e alto acompanhada de gritos.

É interessante o uso de efeitos sonoros no início da criação de tensão. Uma forma de uso eficaz dos mesmos é utilizá-los com uma melodia baixa e calma ou com um ruído que se

intercalam com respiração, passos e silêncio. Um exemplo bom desse tipo de uso para construir tensão é observado no jogo Outlast, onde o personagem anda pelos corredores muitas vezes ouvindo apenas o som dos passos acompanhado do som um pouco ofegante da respiração do personagem, uns gemidos, ranger de portas e correntes se arrastando bem distantes.

Agora que já temos uma base e uma tensão inicialmente construída, precisamos intensificar as coisas. Essa intensificação deve ser feita obrigatoriamente antes de algum evento importante como a revelação do monstro, mas ela também pode ser feita em momentos diversos para criarmos a falsa sensação de segurança. Essa intensificação pode ser feita com a adição de uma cacofonia ou um ruído branco que se intensifica, causando um desconforto e preparando o telespectador para o ápice. É interessante que no momento mais intenso dessa ponte ao ápice seja utilizado um momento de silêncio, isso causa certa confusão no telespectador que estava esperando um momento muito intenso.

Em Pânico (Scream) de 1996 quando Sidney se tranca em uma viatura e tenta utilizar o rádio do carro mas o assassino a impede, vemos o momento ideal de utilizar uma trilha sonora agitada com diversos instrumentos tocando juntos em um ritmo acelerado e alto acompanhada de gritos, perfeito para o ápice.



Visão

Em relação ao visual podemos seguir duas abordagens dependendo do sentimento que queremos causar. Se for para construirmos tensão e abusar da imaginação do telespectador, utilizaremos algum material para atrapalhar a visão, retirando informações. Esse material pode ser sombras, névoas ou pode ser a informação de que o personagem do filme ou o personagem que você controla no jogo não pode ver os seus monstros, pois eles são invisíveis.

Dessa forma, o senso de ameaça está sempre presente e a tensão pode ser criada. Então em uma frase, podemos dizer que a regra é “Não mostre a ameaça real”, pelo menos não de início.

Mude algo assim que o telespectador tenha visto, pode ser a hora de um relógio, uma pintura em um quadro ou até uma bola que estava no ambiente e de repente some. Tudo isso faz surgir dúvidas sobre o que está acontecendo, é a sensação de estranheza sendo explorada visualmente. Sobre não mostrar a ameaça real de início, ela tem um momento em que deve ser apresentada, pois se não mostrar o telespectador se sentirá seguro e perderá o medo, percebendo que não existe ameaça alguma.

Então a ideia é fazer o telespectador se sentir seguro e depois retirar isso dele. Apresente objetos se comportando de maneira estranha sem

explicação, após algum tempo forneça meios que permitam o telespectador entender porque aquilo aconteceu e assim ele se sentirá seguro. Repita o processo, mas dessa vez mostre que a ameaça é real e a partir daí qualquer sensação de segurança será transformada em tensão.

Se a intenção for causar repulsa, nojo, abominação e chocar, podemos mostrar cenas extremamente violentas, com muito sangue, vísceras e restos mortais de humanos ou animais. Essa segunda forma de abordar a parte visual é conhecida como Gore. Geralmente ela é utilizada após a construção da tensão, no momento ápice.

Essas duas formas de abordar a parte visual do terror as vezes parecem ser antagônicas mas se utilizadas de forma dosada e complementar, podemos explorar os diversos espectros do medo.



Tato

E além do audiovisual, algumas mídias interativas, como jogos mobile e de consoles, adicionam uma outra dimensão ao terror utilizando um negocinho chamado haptic feedbacks ou respostas hapticas. Que nada mais são que vibrações no controle ou celular. Essa vibrações aprimoraram a experiência de interação com as interfaces na tela. E são comumente utilizadas como mais uma forma de feedback após o efetueamento de uma importante ação, por exemplo.

Mas, também podem ser utilizados para construir tensão, imagine sentir o batimento cardíaco de um animal que o personagem

do jogo está segurando; ou servir de mais um estímulo sensorial na hora do susto.

Alguns dos melhores jogos para consoles e mobile utilizam desta técnica para tirar melhor proveito de mais um dos sentidos humanos. É o caso de God of War que desde os primeiros jogos da franquia faz o bom uso de vibrações do controle para adicionar mais um feedback do que está acontecendo ao jogador. Até em cinematics a franquia faz uso dessa funcionalidade para incrementar a imersão do personagem. Já no lado do mobile, temos o Swipe Casters que faz o celular vibrar cada vez que passamos por um dos pontos do desenho que precisamos formar.

**FAZEMOS COM QUE O MEDO
ULTRAPASSE A BARREIRA
DA MÍDIA E FIQUE PRESENTE
NA VIDA REAL**

O terror sai da tela

Existem diversos subgêneros do terror: Psicológico, Cósmico, Trash, Found Footage, Gore, Body Horror, Slasher, Sobrenatural, Thriller etc. Cada um deles possui uma especificidade, mas as bases apresentadas nesse texto podem ser utilizadas e percebidas em todos eles, mudando principalmente a intensidade com que são utilizadas. Além do ritmo, da audição, do visual e do tátil, uma boa peça de terror precisa de uma boa história, personagens que gerem empatia e situações que sejam de alguma forma comuns aos telespectadores. Gerando identidade fazemos com que o medo ultrapasse a barreira da mídia e fique presente na vida real, fazendo com que os telespectadores fiquem apreensivos ao passar por um corredor escuro ou escutar algum som estranho quando estiver sozinho em casa. Quando for beber água hoje de madrugada, lembre de considerar se vale mesmo a pena correr o risco e se considerar que vale, tenha certeza de verificar todos os cantos da casa, você pode não estar sozinho. ❶





Memo

Pesquise e salve palavras em inglês e
deixe que o app te mostre diariamente
quais precisam ser revisadas

Baixe gratuitamente na App Store e nos
dê Feedback sobre sua experiência, desta
forma poderemos entregar um app cada
vez melhor para você!



CÓDIGO {

Casa inteligente com ESP e Homekit

TEXTO *Leonardo*
EDIÇÃO *José Mateus*
DESIGN *Carolina*

Vamos começar por partes, primeiro o que é um ESP? É uma placa de desenvolvimento e produção, que possui um microcontrolador com a capacidade de conexão WI-FI já integrada, sem necessidade de nenhum módulo externo para conectar com redes Wireless. O WI-FI é uma tecnologia de rede sem fio que permite que MacBooks, iPhones e outros dispositivos, como o ESP se conectem à internet e assim também permitido que muitos dispositivos troquem informações entre si, criando uma rede capaz de trocar informações e assim podendo controlar o ESP por meio do HomeKit.

E o que é o HomeKit? HomeKit é um framework desenvolvido pela Apple que torna seus dispositivos, como iPhone, Apple TV e Apple Watch, capazes de se comunicar e controlarem acessórios na sua casa. O framework permite desde usar a Siri para controlar os acessórios, usando comandos de voz nativo do sistema IOS. Também existe a possibilidade de criar ações, nas quais é possível definir horários de funcionamento de acessórios e controlar vários dispositivos com um único comando. A Apple possui um aplicativo próprio que é capaz de fazer o controle e configuração dos acessórios, o aplicativo Home, todos devem tá cadastrados e configurados nesse aplicativo, para que seja possível utilizar, mas cada fabricante

pode ter seu aplicativo próprio com controles diferentes dos já oferecidos pela Apple.

Com tudo, esse sistema era fechado até o final de 2019, no qual a Apple liberou parte do HomeKit Accessory Development Kit (ADK), que é o Firmware responsável por fazer a comunicação entre os sistemas embarcados com o Framework do HomeKit. Esse ADK é disponibilizado integralmente por meio do Programa MFI. Ao participar do Programa MFI, você obtém especificações técnicas, componentes de hardware, ferramentas de certificação e o que é de mais moderno em tecnologias Apple para que seja possível desenvolver e fabricar acessórios eletrônicos que se conectem a produtos Apple, usando tecnologias licenciado. Com essa liberação de parte do ADK, qualquer pessoa pode desenvolver um dispositivo usando HomeKit, mas a comercialização é proibida.

É muito simples conectar um dispositivo ao Apple HomeKit. Após configurar o dispositivo smart no aplicativo proprietário, você pode só entrar no Aplicativo Home e escanear o QR Code e o seu dispositivo vai ser adicionado. Quando se usa o ESP, a gente já configura a rede WI-FI pelo firmware. Portanto ao ligar o ESP geralmente já aparece pra ser adicionado automaticamente, caso não, é só usar o código para adicionar no Home.

1. Escolhe uma placa de desenvolvimento capaz de se comunicar com a rede e os componentes que voce se interessa como um Rele. As placas de desenvolvimento mais conhecidos são Esp8266 e Esp32. Esses são mais usados devido ja possuírem a parte de Wifi integrado.

2. Compilar o ADK do HomeKit disponível do GitHub da própria Apple, lá eles explicam passo a passo

3. Escolher uma IDE capaz de compilar e gravar no Sistema Embarcado escolhido, geralmente se usa a IDE Arduino, devido a praticidade e ser capaz de se comunicar com muitas placas de desenvolvimento

4. Escrever seu programa com as configurações necessárias pelo ADK do HomeKit, que se representa principalmente pela criação do acessório, definindo suas características e o que ele é capaz de fazer e o servidor local do acessório, como segue exemplo.

5. Depois disso é so gravar e aproveitar a praticidade de ter um acessório smart para sua casa usando seu iPhone.

Esse breve tutorial serve para mostrar como todos são capazes de ter seu próprio dispositivo smart em casa, pagando pouco e com integração de software. Atualmente o número de exemplos na internet vem crescendo e assim ajuda na hora de escrever seu próprio código. ❶

```
homekit_accessory_t *accessories[] = {
    HOMEKIT_ACCESSORY(.id=1, .category=homekit_accessory_category_lightbulb, .services=(homekit_service_t*[]){
        HOMEKIT_SERVICE(ACCESSORY_INFORMATION, .characteristics=(homekit_characteristic_t*[]){
            HOMEKIT_CHARACTERISTIC(NAME, "Sample LED"),
            HOMEKIT_CHARACTERISTIC(MANUFACTURER, "HaPK"),
            HOMEKIT_CHARACTERISTIC(SERIAL_NUMBER, "037A2BABF19D"),
            HOMEKIT_CHARACTERISTIC(MODEL, "MyLED"),
            HOMEKIT_CHARACTERISTIC(FIRMWARE_REVISION, "0.1"),
            HOMEKIT_CHARACTERISTIC(IDENTIFY, led_identify),
            NULL
        }),
        HOMEKIT_SERVICE(LIGHTBULB, .primary=true, .characteristics=(homekit_characteristic_t*[]){
            HOMEKIT_CHARACTERISTIC(NAME, "Sample LED"),
            HOMEKIT_CHARACTERISTIC(
                ON, false,
                .getter=led_on_get,
                .setter=led_on_set
            ),
            NULL
        }),
        NULL
    }),
    NULL
};

homekit_server_config_t config = {
    .accessories = accessories,
    .password = "111-11-111"
};
```


Letrando ABC: Alfabetização com Realidade Aumentada

TEXTO Kellyane, Lidiane,
Patricia e Ronaldo

EDIÇÃO Beatriz

ILUSTRAÇÃO José João em conjunto
com os assets do app

Quando pensamos em alfabetização estamos falando da aquisição cognitiva básica da habilidade de escrita e leitura. Porém, esse conceito foi sendo progressivamente ampliado em razão de necessidades sociais e políticas. Atualmente, alfabetizado é aquele que sabe usar a leitura e a escrita para exercerem suas práticas sociais.

Observamos que durante a pandemia do Covid-19, um desafio foi lançado nas mãos de educadores e pais no que se trata de educação. A alfabetização, um momento tão fundamental na vida educacional de uma criança, está sendo extremamente prejudicada. Percebendo as dificuldades devido a pandemia e de forma geral do próprio processo de alfabetizar, resolvemos escolher a alfabetização como tema principal de nossa aplicação.

A ciência por trás do Letrando

É comum que o processo de alfabetização ocorra quando o indivíduo está iniciando a sua vida. Segundo o Instituto NeuroSaber, entre 2 e 3 anos as crianças já são estimuladas por cores, sons e letras, e aos 6 e 7 anos, a criança já está pronta e preparada para ser alfabetizada. Entretanto sabemos que esse processo também pode ocorrer na fase adulta.

Fizemos algumas pesquisas sobre a alfabetização em diferentes idades e optamos por trabalhar com o público alvo na faixa etária de 4 á 7 anos.

Nossas pesquisas também foram amplas em termos de acessibilidade, pois queríamos englobar o maior número possível de facilidades para incluir mais crianças. Portanto, estudamos e verificamos as componentes essenciais para os processos de aprendizagem da criança, uma vez que tais processos podem ser facilitados pela interação sujeito-sujeito, onde os pares ou modelos exercem grandes influências. Implementamos filtros de cor e o método fônico sempre dando a possibilidade de desativar o recurso caso ocorra irritabilidade ou resistência por parte da criança. A intenção sempre será de um processo fluído e prazeroso para criança.

Tecnologia, jogos e educação

A chegada da tecnologia digital possibilitou muitas mudanças ao longo do tempo e têm modificado o ambiente educacional. Através de plataformas e jogos digitais abriu-se portas para uma aprendizagem diferenciada que motiva e se aproxima da vida da criança.

Os jogos digitais têm um papel fundamental durante o processo de alfabetização. Ao mesmo tempo em que eles conseguem de-

envolver a concentração, o raciocínio lógico e a colaboração entre as crianças, também incentivam a leitura e a escrita.

A tecnologia pode ser usada para incentivar o aprendizado de forma lúdica, permitindo estratégias mais atraentes e inovadoras por parte dos educadores, trazendo certa autonomia na hora dos estudos para o aluno e aproximando o universo acadêmico do universo do estudante.

Nisto entra a Realidade Aumentada, que diferente do que muitos pensam, não é o mesmo que realidade virtual. A Realidade Aumentada permite a interação entre o ambiente virtual e o mundo físico, fazendo com que estejamos atentos ao mundo real, enquanto nossas possibilidades são expandidas. Essa tecnologia traz uma experiência diferente unida com a sensação de imersão, o que é de grande ajuda quando unido aos jogos educativos, fazendo com que durante o aprendizado, a criança receba estímulos psicomotores através da interação com o ambiente.

Durante o desenvolvimento, nossa proposta também foi tornar acessível o aprendizado em qualquer ambiente. Com a realidade aumentada, foi possível arquitetar um projeto em que a criança poderia usá-lo como ferramenta também em casa ou em ambientes diversos, ajudando pais e educadores neste

momento de educação à distância.

As crianças de hoje não têm dificuldade em utilizar a tecnologia e é relativamente fácil unir os mundos online e offline, o que pode ser muito útil quando se trata do aprendizado.

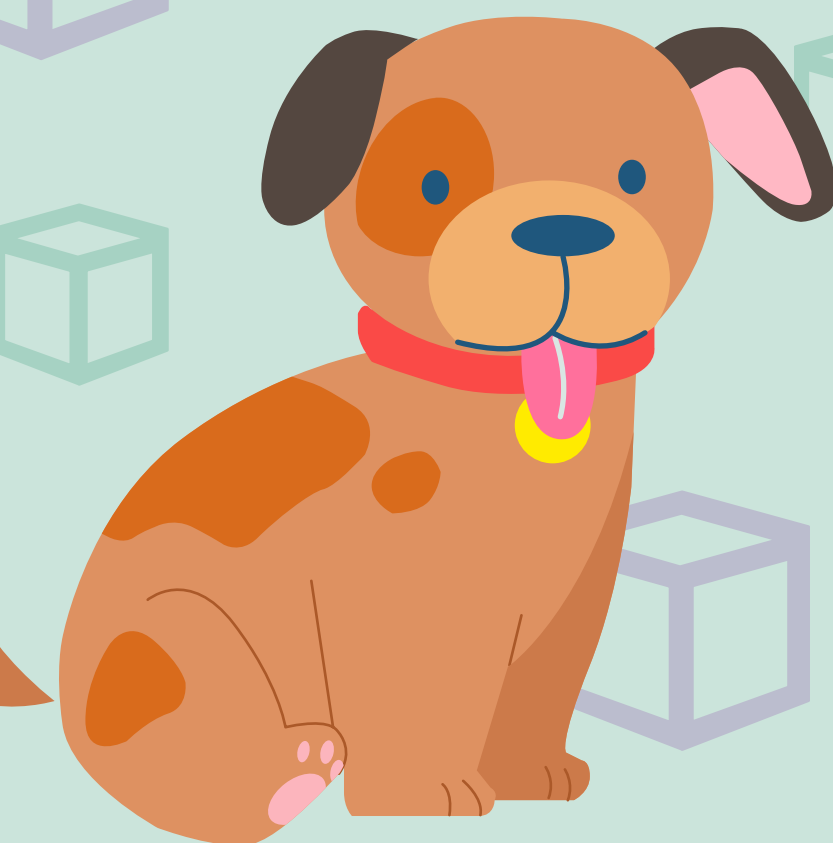
O método sintético

Existem diferentes formas de alfabetizar e cada uma delas destaca um aspecto do aprendizado. Dentre os diferentes métodos existentes atualmente, o Letrando aborda o método sintético.

O método sintético combina som e grafia, oral e escrito através do aprendizado de cada letra, sílaba e palavra individualmente.

O método sintético pode ser dividido em três tipos: Alfabético, fônico e silábico. No alfabético, inicia-se com o aprendizado das letras para formar sílabas e em seguida as palavras. No método fônico, o aprendizado leva em consideração o som das letras e depois une sons de vogais e consoantes para formar sílabas. Já o método silábico, parte do aprendizado das sílabas para formar palavras.

O Letrando aborda esse método ao apresentar para a criança as letras e fazer com que ela encontre no ambiente a letra correspondente. O aprendizado parte do reconhecimento da letra através da sua forma. A letra é apresentada com



No Letrando o aprendizado parte do reconhecimento da letra através da sua forma e da repetição, usando como principal interface o ambiente real por meio da Realidade Aumentada

seu respectivo som, fazendo com que a criança possa relacionar o som com a letra. No jogo a criança consegue formar uma palavra apenas identificando letras e no fim é apresentada a palavra formada com sua respectiva pronúncia.

Esse aprendizado é feito de forma mecânica e a criança adquire vocabulário a partir da repetição.

Colocando tudo em prática

Até o momento, conseguimos ver com clareza as funcionalidades e o objetivo do nosso aplicativo. No entanto, como faríamos para implementar?

Um das dificuldades foi encontrar uma arquitetura que unisse os frameworks ARKit e SceneKit, embora suas implementações não fossem tão difíceis, pois no ARKit conseguimos mostrar elementos no ambiente com poucas linhas de código.

Mesmo com várias provas de conceito que fizemos anteriormente sobre eles, foi diferente quando tivemos que implementar realmente no aplicativo. Um dos pontos-chave foi a tentativa de fazer testes unitários de classes com estes componentes, pois algumas funções envolviam cálculos matemáticos de posições e isso precisava ser testado. Conseguimos componentizar e testar classes de Scene, Plane e ARModel que eram chamadas nas controllers necessárias.

Inicialmente, planejamos algumas tarefas, com foco na funcionalidade principal: a busca de letras no ambiente. Foi a tarefa mais longa e complicada, porém isso era esperado. Durante muitos momentos pensamos que não conse-

guiríamos ter um produto para entregar e ainda entregá-lo a tempo, pois aquela funcionalidade era o nosso aplicativo, sem ela não existia o Letrando. Enquanto isso, tínhamos letras que não apareciam, letras que apareciam mais logo sumiam ou letras com um movimento aleatório e inesperado.

O fato é que trabalhar com ambiente real e perspectivas traz a necessidade de entender alguns conceitos sobre posicionamentos e visão tridimensional. O que levou um tempo até acertarmos. Foram dias complicados, porém conseguimos implementar essa funcionalidade a tempo de lançar uma primeira versão para o Testflight. Ele teve vários bugs reportados, como por exemplo, erros de travamentos ao iniciar a tela de realidade aumentada, que nos acompanhou até o lançamento do aplicativo.

Após o fim do challenge, resolvemos tentar publicar na App Store. Ficamos ansiosos, porém tivemos um balde de água fria ao ter o aplicativo rejeitado. O motivo era problemas na declaração de privacidade do usuário e o fato de o aplicativo ter travado no momento da execução. Mas não desistimos! Fomos ao trabalho e buscamos o erro. Achamos que tínhamos resolvido, mandamos uma nova versão, porém, novamente foi reprovado. Dessa vez, o erro era relacionado ao áudio e foi resolvido. Então, depois de mais de um mês de trabalho, finalmente fomos recompensados com nosso projeto na loja. E sim, foi uma sensação maravilhosa de dever cumprido! Você pode conferir na loja este projeto e contribuir conosco nos dando a sua opinião sobre ele. ❶



Baixe gratuitamente na App Store e nos dê Feedback sobre sua experiência, desta forma poderemos entregar um app cada vez melhor para você!





Cangaceira

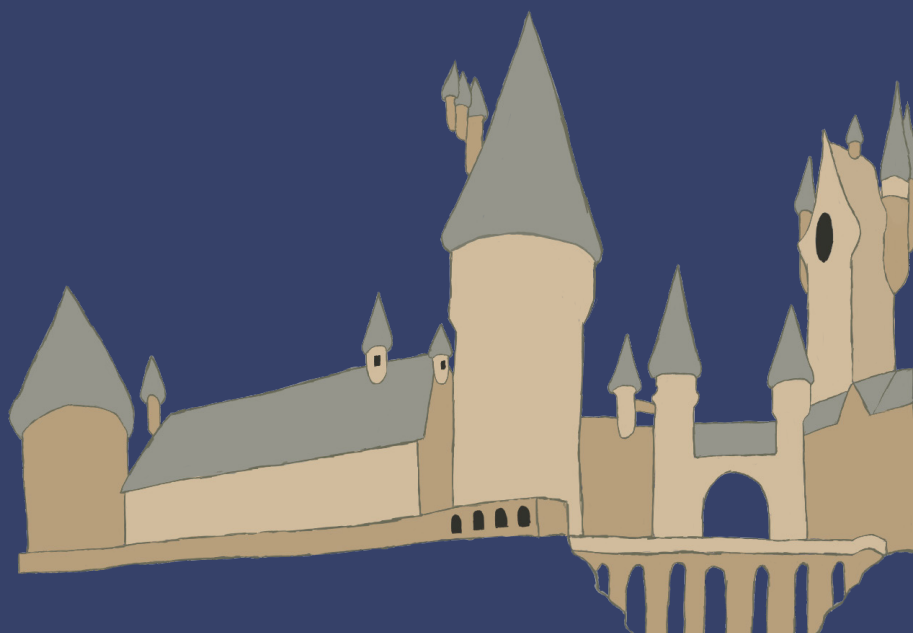
Ajude Virgulina a atravessar sua
jornada em busca de justiça
para o seu povo

Você desenvolve apps? Então isso aqui é
feito para você! Baixe nosso app pelo Github
para estudar o código e nos dar
feedback, assim todos aprendemos juntos!



Desmistificando o time de desenvolvimento SCRUM

TEXTO *Amanda*
EDIÇÃO *Carolina e Kellyane*
ILUSTRAÇÃO *Felipe*



Que a tecnologia tem avançado em velocidade vertiginosa todo mundo já sabe. Para acompanhar essas mudanças, surgiram dezenas de formas de criar e gerir projetos e times de desenvolvimento de software. Algumas empresas tinham vários departamentos separados e um gerente de projetos com uma lista de requisitos fixos trazida do cliente e uma entrega única a ser feita. Nada contra, faz sentido para alguns projetos. Acontece que projetos grandes demoravam, falhavam e custavam muito. Aí veio o Scrum.

Essa ferramenta, que implementa a metodologia Ágil foi muito difundida nos últimos tempos. É baseada em requisitos dinâmicos, ciclos incrementais e pequenas entregas frequentes que geram feedbacks e valor para os clientes. Faz todo sentido com a velocidade que os requisitos mudam.

Como é organizado um time de desenvolvimento no Scrum?

O time de desenvolvimento é operacional: a força tática que transforma tecnologia em

produtos e serviços inovadores. São autogerenciáveis e multifuncionais, divididos em áreas de conhecimento, e não departamentos. Um time deve ser pequeno o suficiente para se manter ágil e grande o suficiente para completar um trabalho significativo dentro da Sprint. Pode conter especialistas em design, desenvolvedores, dev ops, QAs, o que for necessário para completar o trabalho. PMs, POs e Scrum Masters não contam - a não ser que eles botem a mão na massa.

“Não existe hierarquia de comando-controle. O comando é o objetivo do projeto e o controle é sutil”

Mas quem organiza isso?

Esta parte fica com o Gerente de Projeto (ou gestor ou Product Manager). É ele quem permanece liderando e conduzindo a visão, estratégia e requisitos de um produto, facilitando a comunicação interna do time e externa com outras áreas da empresa. Por ter uma visão estratégica, essa pessoa organiza o roadmap do projeto e traz ao time as demandas e expectativas ligadas ao produto.

Espera, esse não é o trabalho do PO?

No Scrum, Product Owner é a pessoa que assegura qualidade do produto e guia o time de desenvolvimento para o caminho correto para as partes interessadas (pessoas que estão pagando, conhecidos como stakeholders). Ela também tira dúvidas acerca do negócio e auxilia o time a quebrar e priorizar requisitos para maximizar o valor das entregas, representando o cliente que nem sempre pode estar lá.

E o tal do Scrum Master?

Essa pessoa é a guia do Scrum, que garante que a metodologia seja bem praticada e que o time seja otimizado e ágil de verdade. Entre treinar a equipe Scrum, modificar o processo e proteger o time, ela acompanha de perto garantindo as boas práticas.

Essa pá de gente em cada projeto?

Lembra que a galera ágil é multifuncional? Nem sempre todas essas cargos estão presentes, mas suas funções sim. É comum o Product Manager assumir o papel de Scrum Master, PO ou os dois. Eles podem evoluir para gerente de programa, cuidando de mais de um projeto relacionado, e para gerente de portfólio, que organiza e prioriza projetos e programas de uma empresa. Muitas nomenclaturas mudam, mas a função básica das lideranças é viabilizar as ferramentas e criar um ambiente propício para o time se organizar, colocar a mão na massa e entregar resultados.

Não existe certo e errado em nenhuma metodologia que se preze. Cada projeto e cada time descobre a cada iteração o que funciona melhor para o próprio contexto. Logo, cabe a essas pessoas, e somente a elas, definir o que deve ser feito para que a entrega seja realizada com sucesso. !

Resumindo quem é quem:

Afinal, esse bando de sigla as vezes pode ser bem confuso...



PM's

Quem lidera e conduz a visão, estratégia e requisitos de um produto



PO's

a pessoa que assegura qualidade do produto e guia o time



Scrum Master

é a guia do Scrum, que garante que a metodologia seja bem praticada



CÓDIGO {

Grand Central Dispatch

TEXTO *Lidiane e José Mateus*

EDIÇÃO *Kellyane*

ILUSTRAÇÃO *Carolina*

Primeiro preciso introduzir dois temas que são de extrema importância para uma melhor compreensão: Simultaneidade e Multithreading.

A simultaneidade ocorre quando várias partes do seu programa são executadas ao mesmo tempo de forma lógica enquanto se comunicam entre si.

Multithreading é uma técnica que usa a CPU do seu dispositivo para fornecer vários threads de execução simultânea. Threads podem ser definidas como tarefas independentes dentro de um mesmo processo que podem ser executadas em paralelo ou de forma intercalada no tempo.

Existem duas principais maneiras pelas quais o iOS permite executar código de forma simultânea: Grand Central Dispatch (GCD) e Operations.

Nesse artigo irei abordar alguns conceitos do GCD que servirão como pontapé inicial para que você possa se aprofundar no tema futuramente.

O que é Grand Central Dispatch (GCD)?

GCD é a implementação da Apple da biblioteca libdispatch da linguagem C. Seu objetivo é gerenciar tarefas que podem ser executados em paralelo, dependendo da disponibilidade de recursos.

Com o Grand Central Dispatch é mais fácil escrever código multithread, independentemente da plataforma ou dispositivo em que seu aplicativo é executado. Ele gerencia um pool de threads compartilhados.

Através dele, é possível adicionar blocos de código ou itens de trabalho para serem despachados em filas e o mesmo decide em qual thread executá-los. O thread usado para executar uma tarefa é controlado pelo próprio GCD, não pelo desenvolvedor.



Os trechos de código são separados em filas com prioridades diferentes para serem despachados!

Síncrono vs. Assíncrono

Você pode despachar uma tarefa para ser executada de forma síncrona ou assíncrona.

Quando uma tarefa é despachada de forma síncrona, o seu aplicativo irá bloquear o loop atual e aguardar até que a execução da tarefa seja finalizada, para depois liberar e deixar disponível a thread novamente para outras tarefas.

Em contrapartida, quando uma tarefa é despachada de forma assíncrona, a tarefa é inicializada porém, o controle retorna à execução do seu aplicativo imediatamente. Assim, uma função assíncrona permite que o seu aplicativo fique livre para executar outras tarefas enquanto a primeira ainda está em execução.

Grand Central Dispatch é como uma grande estação de trem, despachando as pessoas certas nas hora certa.

Fila de despacho

`DispatchQueue`

Uma fila de despacho é uma fila a qual o trabalho pode ser agendado para execução. Você envia unidades de trabalho para esta fila e o GCD as executará em uma ordem FIFO (First In, First Out), garantindo que a primeira tarefa enviada seja a primeira iniciada.

Existem dois tipos de filas de despacho, filas de despacho seriais (Serial queue) e filas de despacho simultâneas (concurrent queue). Uma fila serial tem um único thread associado a ela e permite que apenas uma única tarefa seja executada a qualquer momento. Isso significa que, se duas tarefas são enviadas para uma fila serial, a segunda tarefa precisa esperar até que a execução da primeira seja concluída para que ela seja inicializada.

Já as filas simultâneas são capazes de utilizar tantos threads quanto os recursos do sistema permitirem. Isso permite que várias tarefas sejam executadas ao mesmo tempo.

Fila principal

`DispatchQueue.main`

Quando você cria um aplicativo, o sistema cria automaticamente uma fila principal e a associa ao thread principal de seu aplicativo.

A fila principal é uma fila serial, logo se você executar tarefas de longa duração, ela pode apresentar travamentos ou comportamentos inesperados.

Como regra todas as atividades de interface do usuário devem ser feitas no thread principal, caso contrário, o usuário experimenta atrasos no aplicativo. Por exemplo, se você faz uma chamada de rede para realizar login de um usuário e deseja leva-lo à próxima tela, a chamada de rede deve estar em um thread de segundo plano, mas a transição do usuário para a próxima tela deve estar no thread principal.

Para executar código no thread principal o GCD fornece o `DispatchQueue.main.sync`, e o `DispatchQueue.main.async`. Quando chamamos o `main.async` o thread principal não será bloqueado. Sua tarefa será executada em um thread de segundo plano, e uma vez concluído o thread principal é atualizado. No entanto, ao chamar o `main.sync`, você corre o risco de o thread principal ser bloqueado, pois ele precisa esperar até que a tarefa seja executada.



Filas Globais

`DispatchQueue.global`

Uma fila global é uma fila simultânea que o sistema operacional fornece para que o aplicativo execute tarefas com base na prioridade da tarefa. A prioridade de uma tarefa descreve a sua importância.

A forma como configuramos a importância de uma tarefa é utilizando a Qualidade de Serviço, mais conhecida como QoS (Quality of Service). O sistema operacional usa a QoS de uma fila para determinar quando o trabalho realizado nessa fila é executado. A fila principal possui a prioridade mais alta.

Como o trabalho de prioridade mais alta é executado mais rapidamente e com mais recursos do que o trabalho de prioridade mais baixa, normalmente requer mais energia do que o trabalho de prioridade mais baixa. Especificar com precisão as classes de QoS adequadas para o trabalho que seu aplicativo executa garante que ele seja responsivo e economize energia.

Os atributos de qualidade de serviço, em ordem de prioridade, são:

Atributos de qualidade de serviço:

`DispatchQoS.userInteractive`

Tarefas que tem a qualidade de serviço como `userInteractive` são executadas na thread principal. As tarefas interativas do usuário têm a prioridade mais alta no sistema, sua execução é imediata. Use essa opção para animações, tratamento de eventos ou atualizações na interface do usuário do seu aplicativo.

`DispatchQoS.userInitiated`

Usado para executar tarefas iniciadas pelo usuário, para as quais ele está aguardando resultados. Por exemplo, carregar conteúdo de um email que deseja exibir para o usuário.

`DispatchQoS.default`

Classe de qualidade de serviço padrão, não possui uma prioridade especificada e não é necessário atribuí-la.

`DispatchQoS.utility`

Usado quando você tem tarefa que não impedem o usuário de continuar de usar seu aplicativo. Tarefas para as quais o usuário está disposto a esperar e normalmente possuem um indicador de progresso.

Tarefas como chamadas rede, cálculos, entrada e saída de dados podem ser feitas usando essa classe.

`DispatchQoS.background`

Tarefas em segundo plano possuem a prioridade mais baixa de todas as tarefas. Atribua essa classe a tarefas ou filas de despacho que você usa para realizar o trabalho enquanto seu aplicativo está sendo executado em segundo plano. Tarefas que duram muito tempo e as quais o usuário não está diretamente ciente como backups e sincronização de servidores remotos.

`DispatchQoS.unspecified`

Essa classe representa a ausência de uma classe de qualidade de serviço. Essa opção existe para dar suporte a APIs legadas que podem desativar a qualidade de serviço.



DispatchSemaphore

Em muitas aplicações, a requisição a um determinado recurso é feita por várias threads do mesmo processo. E muitos recursos não podem ser acessados por vários processos ao mesmo tempo. Então, quando um processo A requisitar um recurso e este recurso estiver em posse de outro processo B, o processo A deve esperar o processo B liberar o recurso.

Para evitar que o processo em espera fique consumindo processamento, o matemático holandês E. Dijkstra foi criou um mecanismo chamado semáforo.

O framework Dispatch tem uma classe chamada DispatchSemaphore que implementa o mecanismo do semáforo. Iremos visualizar o funcionamento de semáforo juntamente com a explanação dessa classe.

Imagine várias crianças brincando com bolas e um cesto (com capacidade C) para guardar as bolas. As crianças podem começar a brincar, já estando na posse de uma bola, ou podem pegar a bola no cesto. Elas passam um tempo T brincando e, quando terminam de brincar, guardam a bola no cesto, passando a descansar por um tempo D. Depois retornam a brincar e, assim, sucessivamente.

Para representar esse problema real por meio de um software, teremos que tratar problemas de sincronização, que são:

1) Um deles é quando a criança vai brincar, porém no cesto não existe bola. Para tratar essa situação iremos ter um semáforo chamado full que fará com que a thread criança fique bloqueada quando tentar brincar e o cesto estiver vazio.

2) O outro problema acontece quando a criança termina de brincar e vai deixar a bola no cesto, porém, o cesto já atingiu a capacidade máxima de bolas. A solução é criar outro semáforo chamado empty que fará com que a thread criança fique bloqueada quando tentar colocar a bola no cesto cheio.





```
class Cesto {  
  
    /* O semáforo full terá seu valor inicial 0, pois o cesto  
       iniciará vazio. */  
    let full = DispatchSemaphore(value: 0)  
  
    /* O semáforo empty terá seu valor inicial C que é a capacidade  
       do cesto. Caso a criança seja criada com bola, após brincar ela poderá  
       guardar a bola no cesto, desde que o valor de empty seja diferente de 0 */  
    let empty = DispatchSemaphore(value: C)  
  
    func pegaBola() {  
        /* O método wait() diminui 1 do valor do semáforo. */  
        full.wait()  
        tiraBola()  
        /* O método signal() adiciona 1 do valor do semáforo. */  
        empty.signal()  
    }  
  
    func devolveBola() {  
        empty.wait()  
        colocaBola()  
        full.signal()  
    }  
}
```

Agora, foquemos apenas na implementação da solução desses problemas:

1) A classe Cesto implementará os métodos `pegaBola()` e `devolveBola()`, e são nesses métodos que ocorrem a manipulação dos dois semáforos explicados anteriormente, sendo que os dois semáforos são atributos da classe. É importante ressaltar que existe um atributo que irá representar a quantidade de bolas presentes no cesto, e esse atributo será modificado pelos métodos `tiraBola()` e `colocaBola()`.

2) A classe Criança terá acesso a classe Cesto, e ela implementará os métodos `brinca()` e `descansa()`.

Acima temos os trechos de código dos métodos `pegaBola()` e `devolveBola()`

Sabemos que problemas do mundo real podem ser tratados e simulados por meio de um sistema computacional, porém, a lógica e os mecanismos que ficam invisíveis aos olhos do usuário desse sistema têm que ser muito bem planejados e estruturados. Tudo isso para que o software forneça as informações requisitadas pelo usuário em tempo satisfatório. E com esse artigo você teve uma introdução a alguns exemplos da utilização do GCD, e possivelmente irá utilizá-lo para otimizar seus softwares para as plataformas da Apple. ❶

DESIGN {

CSI: Case Structuring Investigation

TEXTO *Amanda*
EDIÇÃO *Kellyane*
ILUSTRAÇÃO *David*

O trabalho de um designer muitas vezes é digno de um verdadeiro investigador: buscar pistas, ter percepções a partir de evidências, investigar e chegar em soluções para um desafio. Um estudo de caso tem exatamente esse objetivo: evidenciar o processo entre um problema inicial e sua solução. E assim como não é possível apontar criminosos sem evidências formais, não é ideal mostrar telas bonitas no seu portfólio sem nenhum contexto.

Existem muitas maneiras de estruturar um processo de design para mostrar como você desvendou os mistérios do negócio e este infográfico mostra uma.

1. Observação

Isole a cena do crime, identificando e contextualizando o projeto.

O que é? De quem é, pra quem é e qual a importância dele? Esta será a introdução do seu case.

Defina também o objetivo. Se foi para um cliente, diga qual, o que foi pedido, e o mais importante, por que, qual o principal desafio e a definição de sucesso do projeto. Se não foi para um cliente, descreva como você se deparou com a problemática. Uma breve história pode ajudar o leitor a entender o contexto.



Vale lembrar que diferente da perícia, nenhuma solução única funciona para estudos de caso e design. Tudo depende de você, do seu projeto e do tipo de trabalho que realiza. Projete seus estudos de caso com o público alvo em mente, descreva-os detalhadamente e será um sucesso!

2. Registro e coleta de evidências

Mostre os vestígios que você coletou para resolver o desafio. Descreva as etapas do seu trabalho e quais entregáveis fizeram parte dela. Por exemplo:

Contexto: Percepções importantes vindas da pesquisa (desk research, pesquisa quantitativa), análise de concorrentes (benchmark);

Usuários: Definir e justificar o público-alvo, personas, user journey map, jobs to be done e ferramentas para ajudar a entender o seu público.

3. Análises laboratoriais das evidências

Hora de analisar o corpo... do trabalho. A partir dos primeiros vestígios, evidencie as pistas que você seguiu e formas que usou para desvendar o mistério. Etapas que podem entrar:

Validação: Matriz CSD, pesquisa qualitativa e formas de validar a solução;

Desafio: A hipótese definida e forma de validá-la. Pode ser acompanhada de um value proposition canvas e outros canvas de negócio, e métricas para acompanhamento do projeto;

Solução: Ideação: Crazy eight's, protótipos de baixa e média fidelidade, validações a partir de testes de usabilidade, Design System e outros entregáveis visuais.

4. Conclusão

Por fim, mostre os resultados! Informe o que deu certo, o que deu errado, e se no fim das contas o crime, ou melhor, o desafio, foi desvendado. Aqui entram:

Testes: Resultados dos testes do MVP, sejam dos protótipos ou versão desenvolvida. Defina o que quer medir, como e o resumo dos resultados e principais insights.

Conclusão: Suas percepções, reflexões e próximos passos. **1**

HISTÓRIAS {

LISS: Um espaço seguro para a aprendizagem de inovação

TEXTO *Livia, Ravena, Cristiano Araújo e Higor Oliveira.*

EDIÇÃO *Helaine*

ILUSTRAÇÃO *Carolina e escritores*

Fazemos parte de um projeto de educação que cultiva um novo modelo de aprendizagem, visando formar jovens graduandos para um mercado de trabalho cada vez mais competitivo e demandante na área de desenvolvimento de software mobile. Desde o início, sabíamos que era preciso não apenas preparar esses estudantes para serem mão de obra qualificada, mas também para serem pessoas autônomas e protagonistas das suas trajetórias. O que fazemos na prática é formar inovadores, ou seja, pessoas proativas e ousadas, capazes de transformar com pensamento crítico e criatividade qualquer ambiente onde venham a trabalhar, ou quem sabe, utilizar essas habilidades para a fundação de seu próprio negócio. Esperamos formar estudantes que sejam capazes de encontrar oportunidades de impactar de forma positiva a sociedade, propor soluções inovadoras, engajar outras pessoas nessa iniciativa, comunicar e co-construir estas soluções junto com um time plural e diverso. Nosso propósito é transformar e multiplicar transformadores para inovação.

A formação para a inovação necessita de modelos de aprendizagem que precisam se alinhar ao mundo em que vivemos, cada vez mais dinâmico, colaborativo e transdisciplinar. Imagine se pudessemos vivenciar um ambiente de aprendizagem seguro onde seus participantes possam aprender a inovar, experimentar desafios e utilizar a tecnologia como ferramenta

para mudanças coletivas? É com este intuito que construímos o Learning Innovation Safe Space (LISS). Um ambiente de aprendizagem para inovação que promove segurança emocional e colaboração para que todos possam se sentir confiantes ao enfrentar desafios. Sabemos que não é tão fácil aplicar modelos de educação que desenvolvam habilidades socioemocionais. Por isso, o LISS é um novo ambiente que dá suporte à formação de inovadores, que valoriza a cooperação e o reconhecimento das potências individuais para formar protagonistas que implementam mudanças positivas nos espaços em que estão inseridos.

Um ambiente como este deve estimular o desenvolvimento de habilidades alinhadas aos modelos de educação para o século 21. Esse modelo de aprendizagem envolve pensamento crítico, comunicação e cuidados emocionais que são necessários para se posicionar em um mundo cada vez mais dinâmico, diverso, competitivo e colaborativo. Pensar em uma formação inserida em um contexto de constante mudança, incertezas e novas oportunidades, faz com que a aplicação de métodos e ferramentas não sejam suficientes. Precisamos de um espaço seguro para fomentar novos modelos de educação. Por isso, acreditamos no que Sara Hallerman, Colon Lewis e Brad Dresbach afirmam como compromissos de uma educação para o século 21:

“O papel dos educadores no século XXI deveria ser ajudar cada estudante a aprender a aprender. É inspirador de criatividade, encorajando a colaboração, esperando e recompensando o pensamento crítico, e ensinando às crianças não só como comunicar, mas também o poder de uma comunicação eficaz. Estas são competências que os estudantes precisam de desenvolver para prosperar no local de trabalho dinâmico de hoje e de amanhã.”

Como podemos ver, o aprendizado de habilidades para inovação tem um papel central na educação para o século 21. Juntamente com carreira e vida profissional e domínio de mídia e tecnologia, tudo isso complementando elementos tradicionais da educação como currículo e avaliação. Na nossa visão isto vai ainda mais além, todo o conjunto de aprendizado promove a formação de pessoas mais inovadoras, que estão constantemente aprendendo a aprender, e o ambiente de aprendizado deve abraçar tudo isso. Estas habilidades são fundamentais para a adaptação e crescimento em um ambiente de trabalho moderno. Além disso, o aprendizado orientado à inovação apresenta mudanças importantes para quem está envolvido no ambiente educacional: estudantes, mentores e gestores. A mudança principal é a saída de um paradigma de replicação do que um mentor já sabe para outro de criação, combinação,



busca e aplicação de conhecimentos que vão além daqueles dominados pelos mentores.

O aprendizado de inovação é multidimensional, onde profissionais são formados e desenvolvem quatro habilidades principais ou 4Cs. Primeiro a Criatividade para se pensar fora da caixa e criar soluções diferenciadas para necessidades relevantes em projetos e ações; pensamento Crítico para se analisar alternativas, diversificar e evoluir sempre; Comunicação que nos faz escutar ativamente e entender melhor as outras pessoas, e também compartilhar de forma efetiva nossas ideias e impressões; e finalmente, mas não menos importante, Colaboração que permite fortalecer potências, competências e motivações coletivas tão maravilhosas quanto diversas para que juntos aprendamos a impactar positivamente a nossa sociedade. Os 4Cs nos fazem pensar em uma educação holística e transdisciplinar. Ou seja, uma educação que não reduza os estudantes apenas aos conhecimentos técnicos que precisam adquirir para o mercado de trabalho.

É por entender a complexidade de cada indivíduo e o potencial de sua trajetória que cultivamos uma educação transformadora e alinhada com a formação para o século 21. Mas o que queremos dizer com isso? Transformar é inquietar, tirar da zona de conforto. É conduzir os estudantes e todos que compõem o ambiente educacional para serem agentes ativos na construção de seus conhecimentos e se apropriarem desse processo como fruto da colaboração, fomentando a diversidade e cuidado em comunidade. Acreditamos em um ensino baseado em paixões, propósitos e desafios individuais e coletivos. Mas para que isso aconteça é necessário compreender os obstáculos para o desenvolvimento de um ambiente seguro baseado nessas motivações, que ocorrem quando os estudantes se apropriam de seus aprendizados.

Por que nutrir um ambiente seguro para o aprendizado de inovação?

Ao construir uma educação pautada na promoção da ousadia e a constante busca pela superação de desafios, surgem novas necessidades de ambientes de ensino-aprendizagem e com elas novas dificuldades. Dado o propósito de multiplicar transformadores, as aulas e as atividades educativas precisam enxergar o estudante de forma holística, incluindo também um plano para o desenvolvimento de habilidades intra e interpessoais e atitudes como a valorização das

suas próprias conquistas pessoais. Atitudes que estão relacionadas aos 4Cs para enfrentar problemas coletivamente. Assim, a proposta de romper com as abordagens tradicionais de educação que possuem uma visão mecanicista da prática educacional e são voltadas para a reprodução do conhecimento, faz com que estudantes e mentores tenham que aprender a exercer um papel mais ativo, se permitindo criar, explorar e construir seu próprio aprendizado.

No entanto, o fomento e a manutenção de um ambiente educacional que promova essa aprendizagem ativa, crítica, criativa e autônoma é repleta de desafios, sendo um dos mais frequentes e importantes o fenômeno do impostor. Uma questão onde as pessoas se sentem inseguras em relação à sua competência, uma vez que atribuem o seu sucesso a outros fatores, como a sorte. Isso impacta fortemente o desenvolvimento do conhecimento dos estudantes.

Na nossa experiência em educação, este fenômeno é ampliado quando existem estudantes de vários cursos trabalhando juntos, pois há pessoas com domínios de conhecimento diversos. Por exemplo, em uma equipe transdisciplinar com designers, programadores e estudantes de outros cursos, como o de cinema, pode haver um aumento de insegurança e comparações, por conta da diversidade de competências e conhecimentos. Esse fenômeno pode impactar negativamente a maneira que a pessoa lida com desafios e situações vivenciadas no ambiente de aprendizagem, uma vez que ela não se sente capaz de realizar as atividades de uma maneira tida como satisfatória para ela que está sempre se comparando com os outros. Essa falta de reconhecimento do seu potencial é prejudicial também para a experimentação e a aprendizagem de coisas novas, dado que o medo de falhar pode lhe paralisar frente a novos desafios e até resultar em autossabotagem. Esse problema atinge não só o estudante como também a sua equipe e mentores, constituindo um obstáculo para a evolução do projeto e para o desenvolvimento de todos.

A falta de reconhecimento e valorização de conquistas pessoais impacta também outros desafios como o desenvolvimento da autonomia. Por um lado, a autonomia faz com que o processo de aprendizagem se alinhe com as curiosidades, paixões e propósitos do estudante, contribuindo para a construção de uma trajetória própria. Por outro lado, há o peso de realizar escolhas e guiar a sua aprendizagem, o que pode ocasionar em um

aumento de ansiedade e insegurança, atingindo também os mentores. Como consequência dessa ansiedade e insegurança surgem dúvidas, por exemplo: como desenvolver a autonomia dos estudantes compreendendo o limite de suporte que deve ser ofertado? Como conseguir que os estudantes estabeleçam objetivos de aprendizagem que incluam também seus desejos e não só as necessidades dos projetos? Além disso, como dar suporte para que os estudantes permaneçam fora da zona de conforto, mas ao mesmo tempo seguros e confiantes em se desafiarem?

Todas essas dúvidas e problemas revelam que uma educação voltada para o desenvolvimento de habilidades para o século 21, que propõe uma visão totalizadora, é repleta de desafios que afetam a todos. Isso torna relevante direcionar formação do estudante em seus diversos âmbitos alinhada às necessidades do seu contexto. Como nos mostra Bolzani (2015), a educação holística enxerga o aluno como sujeito autônomo, crítico, participativo e transformador da sociedade. Ele é entendido como um sujeito único, repleto de inteligências, capaz de inovar e de questionar. A complexidade de lidar com os estudantes em sua totalidade faz com que haja a necessidade da construção de ambientes educacionais acolhedores e promotores de colaboração.

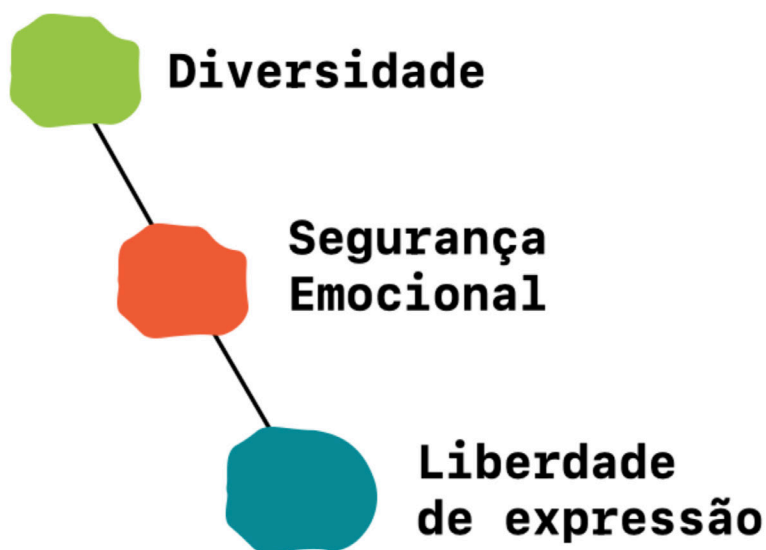
Assim, é fundamental que o ambiente de apren-

dizagem promova suporte material e social, como a segurança emocional e também a confiança para que todos possam ousar sem grandes impactos emocionais. Mas mais do que isso, é preciso sempre mobilizar todos para a busca de forma coletiva da resolução dos seus próprios desafios para ousar e co-criar. É exatamente isso que o LISS se propõe a apoiar, por meio da estruturação de um ambiente que atenda a essas necessidades e anseios.

Como o LISS constrói um Safe Space para o aprendizado de inovação?

A maneira como o LISS é fomentado para transformar pessoas em potenciais inovadores não existiria sem o alicerce de um espaço seguro em que os estudantes possam se desenvolver. Nesse sentido, para pensarmos sobre o LISS é importante discutir sobretudo o que é um Safe Space. Existem várias definições e também cuidados que precisam ser tomados quando se fala de safe spaces. Um safe space é um espaço de construção coletiva. Ele depende fortemente do engajamento das pessoas. Não é algo que basta ser declarado e está construído, ou seja, o Safe Space se concretiza na experiência das pessoas, não podendo ser decretado.

Ambiente Seguro



Portanto, ao desenvolver um Safe Space entendemos que é preciso constituir práticas e experimentações constantes que, naturalmente, garantem erros e acertos no processo, visto que a diversidade de ideias e pontos de vista tornam a sensação de pertencimento a um espaço seguro distinta. Entende-se o Safe Space como um ambiente que é nutrido constantemente de práticas que fortalecem a segurança emocional e o bem-estar coletivo dos envolvidos, onde as pessoas são observadoras ativas de suas necessidades e potencialidades. Além disso, estudiosos da área como Holley e Steiner (2005) pontuam que um Safe Space encoraja a participação e o compartilhamento honesto de ideias, bem como gera um clima que permite que os estudantes se sintam seguros o suficiente para tomar riscos, expressar seus pontos de vista e compartilhar e explorar seus conhecimentos, atitudes e comportamentos. Assim, um Safe Space promove um ambiente que estimula o processo de autodescoberta, em outras palavras, o processo de reconhecimento das suas motivações, posicionamentos e desejos. É por meio da reflexão crítica dos estudantes sobre a sua trajetória pessoal que um Safe Space atua no fortalecimento da autonomia dos estudantes.

Mas como falamos anteriormente, o processo de ser mais autônomo, seguindo as suas motivações e necessidades no aprendizado, gera inseguranças que afetam o emocional dos estudantes e mentores. É por isso que um ambiente de aprendizagem para inovação deve também fornecer segurança emocional.

Apesar dos benefícios que um Safe Space pode proporcionar, também há críticas ao modelo. As principais delas são que quando se implementa um Safe Space leva ao cerceamento da liberdade de expressão e um patrulhamento de comportamentos e conversas para prevenir micro agressões. Existe a preocupação de que o Safe Space encoraje a formação de grupos de estudantes que se isolem de discussões sobre questões que podem tirá-los de sua zona de conforto. Contudo, quando se trata de aprendizagem, são envolvidos processos que naturalmente necessitam que estudantes saiam de sua zona de conforto. Lidar com a pluralidade e o desconhecido não é fácil, mas um Safe Space não deve ser entendido como um ambiente no qual não há conflitos e desconfortos como pontuado por Boostrom (1998):

“Se sentir seguro não é o mesmo que se sentir confortável. Para crescer e aprender, estudantes devem com frequência confrontar questões que trazem desconforto e que forçam a enfrentar suas crenças e questionar quem são.”

Assim, em um Safe Space, alunos e mentores devem se orientar a gerenciar conflitos e não proibi-los. Um ambiente seguro deve promover discussões que envolvam a escuta ativa e a reflexão crítica sobre diversos pontos de vista. O Safe Space é e deve ser um espaço de posições divergentes, mas ao mesmo tempo ser um espaço no qual as pessoas sintam-se protegidas para dialogar de maneira respeitosa e aberta, e aplicar de forma colaborativa essas divergências para construir conhecimentos e soluções diferenciadas. É importante o entendimento que estamos

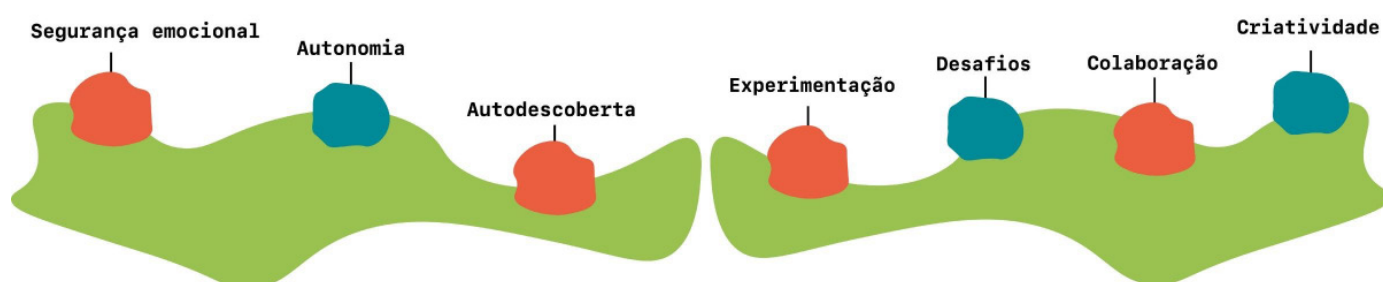
sempre em constante aprendizado e que conflitos fazem parte de um processo de autonomia, ousadia e liberdade de expressão na construção do aprendizado. Agora que entendemos o que é um Safe Space vamos conhecer o LISS.

Afinal, que é um LISS?

Podemos dizer que o LISS é um ambiente seguro voltado para a aprendizagem à inovação. Mais especificamente, é um ambiente que promove a segurança emocional e autonomia para se auto-descobrir na medida em que se

aprende a experimentar, enfrentar desafios e transformar comunidades com colaboração e criatividade. Esse ambiente permite que as pessoas envolvidas recebam apoio e se sintam mais confiantes para traçar a sua própria jornada de aprendizagem. Além disso, é um ambiente que faz com que os ganhos de experimentação e saída da zona de conforto sejam maiores que possíveis desconfortos e impactos negativos, resultando em pessoas mais engajadas em se desenvolver e transformar o seu entorno. Em nossa perspectiva um LISS pode ser retratado como na imagem abaixo.

Learning Innovation Safe Space



Podemos ir ainda mais além. Para inovar no atual contexto histórico e social é necessário, além do aprendizado de competências técnicas, o ganho de habilidades pessoais e interpessoais, como explicitamos anteriormente. Pensando nisso, o LISS propõe 8 categorias de desenvolvimento individual e coletivo, sendo elas: Autodescoberta, Autodesafio, Pensamento Crítico, Comunicação, Divergir e Colaborar, Cuidados Emocionais, Liberdade de Expressão e Criatividade. Podemos ver de forma mais detalhada o que cada categoria significa nas próximas páginas.

Assim, percebemos que estimular a construção de um

ambiente para uma comunidade cada vez mais autônoma e engajada em desafios, onde as pessoas se tornem protagonistas do seu aprendizado e busquem ativamente o desenvolvimento pessoal e coletivo é o que constitui um espaço seguro para aprendizado de inovação. Somos responsáveis por um processo de transformação individual e coletiva, de transformações constantes e de aplicação efetiva da educação para inovação. Vemos o Learning Innovation Safe Space como o ambiente para práticas em inovação com o intuito de formar profissionais capazes de exercer impacto positivo no seu entorno e liderança na im-

plementação de mudanças nos espaços em que estão inseridos. Profissionais com consciência da sua trajetória profissional, que valorizam a cooperação, que reconhecem o potencial de se trabalhar em equipe e que valorizem o cuidado com a comunidade e o acolhimento da diversidade.

Dessa forma, nós que construímos o LISS diariamente, sentimos que fazemos parte de uma experiência de educação e de um ambiente comunitário especial. Contudo, tínhamos muita dificuldade em definir esse ambiente. Falávamos muito de Safe Space, mas não sabíamos exatamente o que era isso. Também tinha-

Autodescoberta

Ter consciência do seu desenvolvimento pessoal através das suas habilidades e interesses.

Pensamento crítico

Explorar, analisar, refletir, questionar e propor através do campo das ideias.

Desafios

Enfrentar seus medos e superar limites ao sair da zona de conforto para aprender.

Criatividade

Se arriscar ao buscar diferentes possibilidades para criar soluções diversas e originais.

Habilidades de Comunicação

Se expressar e ouvir de forma empática visando a co-construção.

Cuidados Emocionais

Ter atitudes acolhedoras que auxiliem no desenvolvimento da comunidade por meio do apoio e do cuidado.

Divergir e Colaborar

Expressar suas contribuições e visões individuais, atuando em conjunto para enfrentar um desafio em comum.

Liberdade de Expressão

Promover a livre exposição de pensamentos, ideias e convicções com respeito, ética e escuta ativa, sem julgamentos morais.

mos consciência de que fazemos parte de um Safe Space para uma educação diferenciada, um ambiente com um propósito de formar transformadores. Assim, procuramos dar a nossa versão do que é um LISS porque não só acreditamos nesse conceito, como temos a convicção de que, apesar dos desafios, estamos vivenciando algo que caminha na direção de um Learning Innovation Safe Space, um espaço seguro para o aprendizado em inovação. Quando refletimos durante a escrita deste artigo nos identificamos muito com

os conceitos que a gente propôs. Passamos horas não só escrevendo mas discutindo e debatendo, e colocando nossas opiniões e visões, muitas vezes divergentes e é claro também se divertindo. Foi uma experiência de auto-descoberta para nós e que queremos compartilhar com vocês. Queremos que esse artigo inspire, gere debates e questionamentos, reflexões e ações para o começo de uma jornada de evolução contínua de um LISS para todos.

Na próxima edição iremos compartilhar a nossa visão de

que já fazemos parte de um LISS em desenvolvimento, que constantemente passa por vários desafios para mentores e estudantes enfrentarem. Iremos dar exemplos do que para nós são evidências de que nosso ambiente é tão importante para quem o compõe e o porquê suas práticas se encaminham para o alinhamento das características de um espaço seguro para a aprendizagem de inovação. Queremos ouvir suas contribuições, fomentar reflexões para possamos pensar juntos e transformar o nosso ambiente de aprendizagem. ❶



Pantry

Pantry

Sua nova forma de organizar sua
despesa doméstica

Baixe gratuitamente na App Store e nos
dê Feedback sobre sua experiência, desta
forma poderemos entregar um app cada
vez melhor para você!



Testes unitários x Singleton

TEXTO *Ronaldo*
EDIÇÃO *Beatriz*
ILUSTRAÇÃO *José João*

O padrão de projeto **Singleton** é uma verdadeira fonte de conflitos no desenvolvimento de softwares, há quem ame e há quem odeie. Ele é um padrão de projeto criacional que gera um objeto com seu acesso compartilhado durante toda a aplicação. Ou seja, cada classe que usa esse objeto tem acesso a mesma unidade de memória em simultâneo. Isso te lembra algo? Variáveis globais! Esta característica tem suas vantagens, como o fato de não precisarmos nos preocupar com instanciação, configuração de parâmetros e veracidade do objeto.

Em geral, o padrão Singleton é a solução mais fácil e rápida. Tudo já vem incluso no singleton, pronto para uso. Incrivelmente ele se torna ruim pelos mesmos motivos. Primeiramente, ele viola o princípio da responsabilidade única, pois sua classe apresenta duas responsabilidades: criar a instância e devolver o objeto. Também deixa o código acoplado a classe singleton, dificultando a customização dela. Ele não permite uma maleabilidade da classe, logo este padrão dificulta a criação de testes unitários.

Testes unitários são a base de um bom código. É a forma de garantir confiabilidade para nossas aplicações em todas as possíveis alternativas cobertas pelo teste. Com a criação de testes unitários é possível descobrir bugs, confirmar comportamentos e documentar

o código. Além disso, ele é de suma importância em projetos de longo prazo. Os testes agilizam a refatoração de classes e funções, pois possibilitam a constante verificação se tal classe alterada ainda funciona da forma esperada.

Para se ter bons testes unitários, eles devem seguir algumas regras. A principal regra é que testes devem ser isolados. Ou seja, cada método deve ser testado isoladamente. Devido a isso, testes garantem uma segurança que o método funciona da forma correta e esperada. E é exatamente neste quesito de independência que as classes singleton atrapalham os testes. Pelo fato do singleton ter uma instância compartilhada, a criação de testes isolados é complicada. Fica difícil aplicar os testes de forma aleatória, pois como a instância é a mesma, qualquer mudança na classe interfere nos outros testes. Além disso, tira a possibilidade de usar mocks de classes. Mocks são dependências criadas especialmente para os testes que visam simular algum comportamento específico de classe. Os mocks são uma saída útil em testes, para simular um comportamento específico de uma classe. Porém, caso a classe seja singleton, não temos como usar de mocks e polimorfismo, pois o construtor é privado e único, impossibilitando criação de classes customizadas.

Mesmo com esses problemas, podemos pensar em possíveis soluções que nos permitam testar classes singleton. A mais comum é usar os métodos padrões de testes, chamados no início e fim de cada rodada de teste. Em Swift, esses métodos são `setUp(Início)` e `tearDown(fim)`. Ao usá-los, podemos instanciar um objeto da classe singleton como variável da classe de teste e inicializar e anular ela a cada ciclo de testes, sempre voltando ao estado base da classe. Dessa forma, todos os métodos de testes usam este mesmo objeto. Isto resolve o problema em parte. O certo é que cada método de teste tenha todo o processo de configuração do objeto nele mesmo, garantindo assim sua independência. Além disso, caso a classe singleton envolva elementos de `save` e `load`, pode ser necessária uma pré-configuração maior antes de cada teste, tornando ele muito complexo e demorado.

TESTES UNITÁRIOS SÃO A BASE DE UM BOM CÓDIGO. É A FORMA DE GARANTIR CONFIABILIDADE PARA NOSSAS APLICAÇÕES EM TODAS AS ALTERNATIVAS COBERTAS PELO TESTE.

O desenvolvedor pode utilizar outras soluções ao invés do Singleton. A injeção de dependência surge como uma delas, pois permite uma customização da classe. Este pattern se torna ótimo para testes, pois é possível injetar classes mocks e com isso realizar testes mais randômicos, específicos e isolados. Para problemas mais complexos, podemos pensar no Factory Pattern.

Factory pattern é um padrão de projeto criacional que cria um objeto em pequenos passos e assim construir um objeto maior e mais complexo. Ele pode ser dividido em dois patterns bem similares, o Factory Method e Abstract Factory. Como sua base de criação é através de protocolos, sua implementação se torna abstrata e de fácil realização de testes, pois mais uma vez podemos usar o conceito de mocks para criar diferentes objetos e testar diferentes possibilidades.

Por fim, não podemos demonizar o Singleton. Ele tem sim, suas utilidades! Linguagens de programação possuem frameworks que utilizam ele, como o Core Data em Swift que fornece um singleton para seu acesso. Singleton é útil em classes que não queremos em múltiplos contextos. Porém, para outros problemas, vale a pena gastar um pouco mais de tempo pensando em soluções mais elaboradas e principalmente que ajudem na realização de testes unitários. ❶

PROCESSOS {

Criatividade e que tipo de habilidade é essa?



TEXTO José João
EDIÇÃO Brena
ILUSTRAÇÃO José João

Quantas vezes na sua vida você se deparou com: “não sou criativo o suficiente”, “não tenho criatividade para resolver este problema”, “estou com bloqueio criativo” ou “não consigo fazer uma solução criativa”? A ideia desse texto é te ajudar a solucionar essas questões de maneira criativa. Exatamente. Resolver problemas de criatividade com criatividade.

É extremamente comum que as pessoas vejam a criatividade como um dom. E essa vai ser minha primeira dica: pense na criatividade como uma skill, uma habilidade. Mas não uma skill que nasce com você e aparece quando quer, e sim uma skill que pode ser treinada e aperfeiçoada. Deixa eu tentar te explicar de um jeito mais técnico.

O Professor de neurociência cognitiva Arne Dietrich possui uma teoria de que o ser humano possui quatro tipos diferentes de criatividade e cada tipo de criatividade depende de uma

região específica do cérebro e de uma condição específica para o seu uso. São elas:

Deliberativo-cognitivo: Caracterizada pela união de informações já existentes no cérebro com informações que estão chegando. É a criatividade mais pragmática e baseada em um conhecimento massivo de alguma área.

Deliberativo-emocional: Caracterizada pela união das informações já existentes no cérebro com as próprias emoções do indivíduo. No fim, é uma criatividade menos lógica e mais subjetiva.

Espontâneo-cognitivo: Caracterizada por aparecer em momentos de relaxamento e descanso. É aquela solução que surge no meio do banho ou no meio de uma caminhada.

Espontâneo-emocional: Caracterizada especialmente por momentos de epifania. Possui



ligação profunda com o lado emocional e não precisa do conteúdo cognitivo para ser ativada.

Baseado nessa teoria, podemos então trabalhar nosso cérebro para alcançar as criatividades deliberativas com mais frequência. Pois elas irão depender dos seus conhecimentos e experiências emocionais já adquiridos. Agora faz um checkpoint mental com essa informação. Vamo ver outra teoria.

Segundo a pesquisa liderada por Roger Beaty, especialista em neurociência cognitiva pela Universidade Harvard, pessoas que possuem ideias originais e criativas no seu dia a dia, possuem o cérebro diferente das outras. Isso porque essas pessoas possuem padrões diferentes em suas conexões neurais. Ou seja, sabendo que o fluxo das suas conexões neurais dependem dos seus conhecimentos prévios, podemos concluir que quanto mais informações um indivíduo consegue coletar ao longo da sua vida, maiores as chances de surgirem ideias inusitadas e únicas.

Então, com isso, aqui vai minha segunda dica: consuma o máximo de culturas e conteúdos diferentes da sua realidade. Todos esses conhecimentos adquiridos vão te ajudar a criar

conexões inusitadas e a pensar de maneiras diferentes

a respeito das novas informações que estão chegando. Dessa maneira, você estará trabalhando bem as suas criatividades deliberativas.

Mas e como trabalhar nossas criatividades espontâneas? Afinal, são espontâneas. Então vamos lá, quantas vezes você conseguiu pensar em uma resposta ótima para dar em uma discussão durante o banho? Ou quantas vezes você descobriu a solução de um desafio, seja de um jogo, seja de um código quando estava apenas caminhando na rua?

As relações dessas e de tantas outras situações semelhantes é que nelas estamos em momentos de descanso. E essa é minha terceira dica: Descanse. E por que? É comprovado que quando o lobo temporal direito tem mais atividade que o lobo frontal, o inconsciente toma conta e a ideia nasce. E para o lobo frontal descansar apenas: descanse. Quando o estresse ou cansaço tomar conta? Se levante, ande um pouco, tome uma água e tente descansar.

E falando em descanso, trago a quarta dica: ande sempre com um bloco de papel e caneta.

Você nunca saberá quando sua criatividade espontânea irá atuar, então é importante que você ande sempre com papel e caneta por perto, dessa maneira, quando você sair do descanso você poderá trabalhar de fato na tua ideia.

E é sobre trabalhar que vem a minha quinta dica: invista tempo no que você quer ficar bom. De acordo com pesquisas, em média, precisamos de 10 mil horas de prática para ficarmos excelentes em uma atividade específica.. E quando uma pessoa consegue atingir esse “estado de excelência” é quando os trabalhos diferenciados surgem. Beethoven, por exemplo, começou seus estudos em música com apenas 5 anos e aos 8 anos já participava de recitais na Academia de Sternengass e já era tratado como gênio. E bem, onde o resto da história deu, você já sabe.

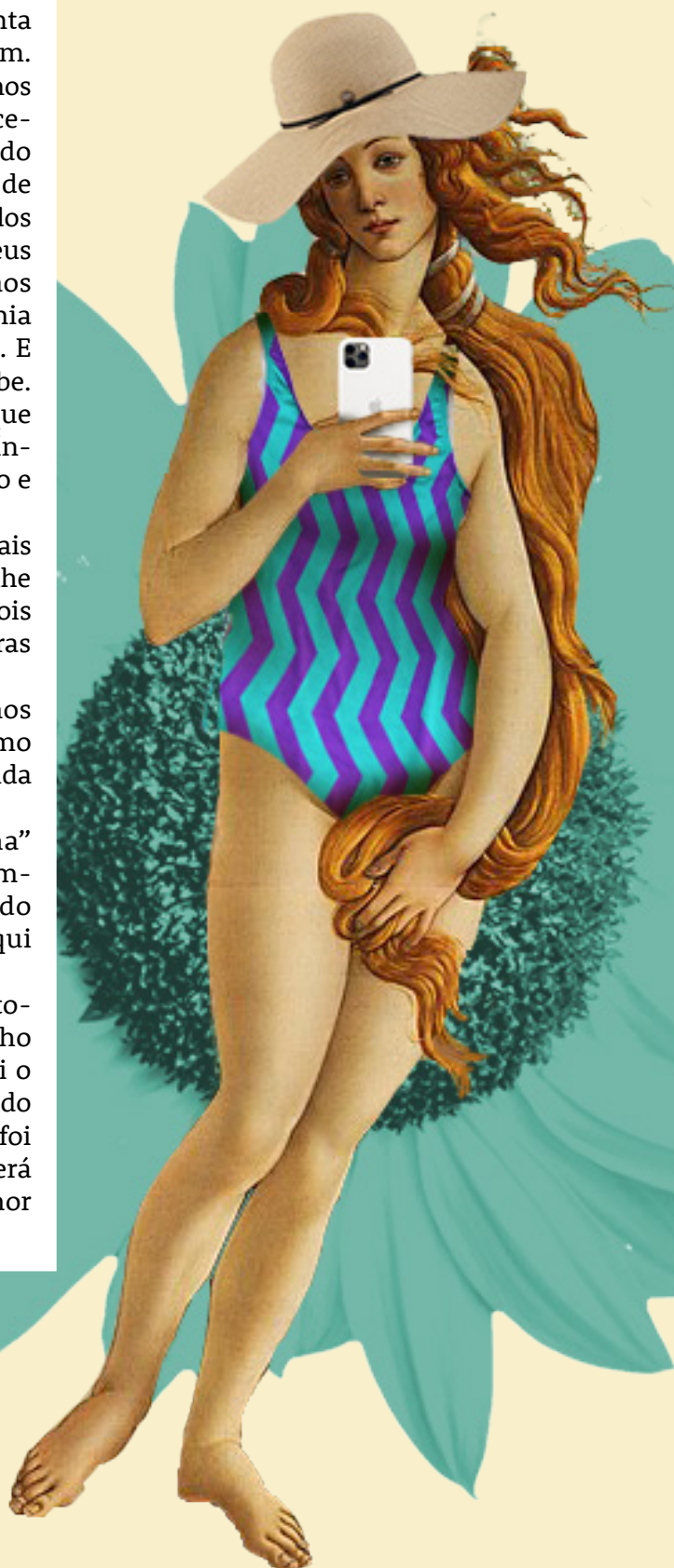
Por fim, e é a partir dos seus trabalhos que surge minha sexta e última dica: fuja da síndrome do impostor. E fuja no início, no meio e no fim do seu processo criativo.

O início, é o momento que você se torna mais propício a desistir de uma ideia, então trabalhe nela apenas com você mesmo. E apenas depois de amadurecer a ideia, compartilhe com outras pessoas, caso seja seu desejo.

O meio é o momento onde mais criticamos o que estamos produzindo. Em “Roube como um artista”, Austin Kleon nos apresenta “A vida de um projeto”

E o momento de “noite negra da alma” sempre chega para todos nós. E é super importante que esse momento seja encarado como “ainda não terminei, então esse aqui não é o resultado final”.

O final é o momento mais subjetivo de todos. Quantas vezes olhamos para o trabalho incrível de alguém e a pessoa fala “eu odiei o que eu fiz”. O momento final deve ser encarado como um momento de reflexão sobre o que foi feito, o que pode melhorar, assim você poderá melhorar o seu processo e deixá-lo do melhor jeito para você. ❶





Tempo Com Deus

Tenha a Bíblia em diferentes versões, com um design simples e acessível, e espaço reservado para anotações.

Baixe gratuitamente na App Store e nos dê Feedback sobre sua experiência, desta forma poderemos entregar um app cada vez melhor para você!



CÓDIGO {

A importância da Matemática para desenvolvedores

TEXTO *Ronaldo*

EDIÇÃO *Mateus Rodrigues*

ILUSTRAÇÃO *David*



Não há como falar de computação sem falar de matemática pois a história de ambas possuem uma grande interseção. O primeiro computador foi imaginado por Charles Babage por volta de 1820. Sua ideia foi motivada pela automatização de cálculos logarítmicos. Para isso, ele pensou em uma máquina ou dispositivo que realizasse esse processo de forma sequencial e rápida. Isto foi o embrião do computador, e ao longo dos anos outros grandes cientistas matemáticos foram aperfeiçoando essa ideia, até a chegada de Alan Turing, mais de 100 anos depois, considerado o pai da computação e um matemático excepcional. Turing criou uma máquina movida à instruções e com funcionamento interno fundamentado em cálculos matemáticos binários e instruções lógicas, com a capacidade de “quebrar” a criptografia utilizada pelos nazistas na 2ª guerra mundial. Após esses acontecimentos, foi perceptível a evolução da computação, e todas as áreas integradas à ela, sendo uma delas a matemática. Resumindo: Computação é matemática em sua essência.

*Computação é
matemática em
sua essência.*

$$(X) \int_0^{\infty} \sum_{i=0}^{\infty} \frac{A(x)}{2\pi} \quad \sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}} \quad E = mc^2$$

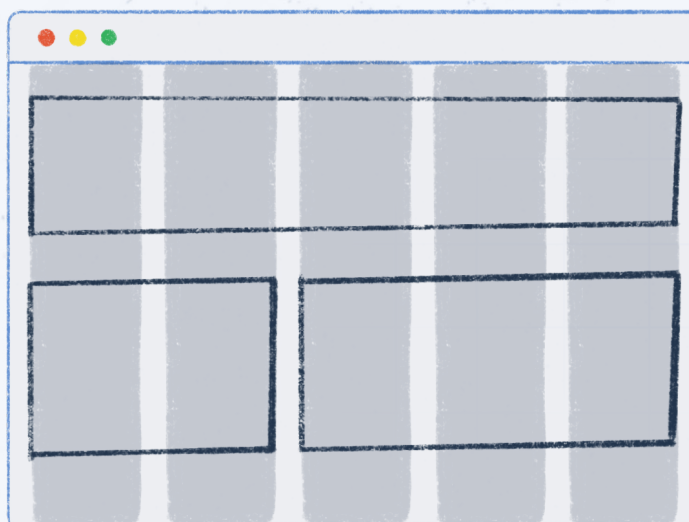
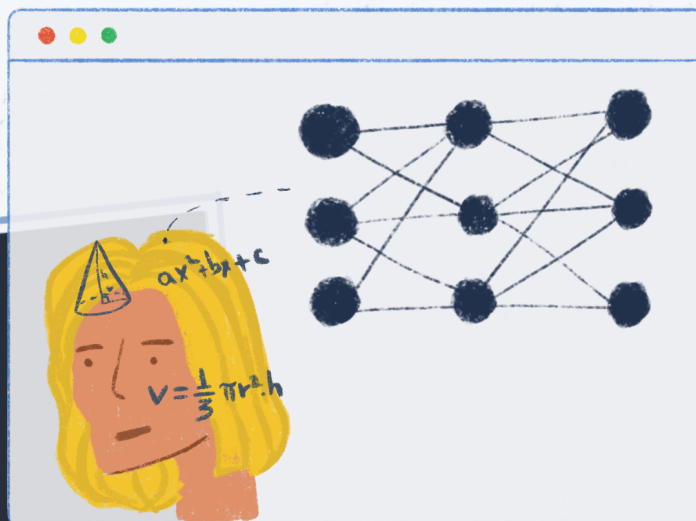
$$V - E + F = 2$$

A grande importância da matemática não é decorar fórmulas, mas sim na abstração, motivação e pensamento lógico.

Matemática em cursos de graduação

Atualmente a computação se tornou um tema de muitos estudos e aplicações. Sua influência está em toda a sociedade, por meio de computadores, smartphones ou qualquer aparelho eletrônico. Com isso, foram criados vários cursos que envolvem o estudo da computação em suas diversas áreas como: Ciência da computação, Engenharias, Sistemas e Mídias Digitais, Análise de Sistemas, Redes de Computadores entre outras, sendo que alguns desses necessitam de um embasamento teórico maior da matemática. Entretanto, em geral ela não é bem aceita pelos alunos, que não enxergam aplicações práticas dos cálculos em seus trabalhos. E em parte, isso é verdade, o uso dela algumas vezes não acontece.

A grande importância da matemática não é decorar fórmulas ou calcular funções na calculadora, mas sim na abstração, motivação e pensamento lógico que justifica e prova que tal ideia é correta. Isto que deveria ser aprendido e aplicado em projetos práticos da computação. Os cursos acabam não ajudando nesse sentido. Em grande parte as cadeiras de matemática se distinguem de aulas de desenvolvimento, não havendo uma integração entre elas, algo que tente ligá-las. O que acontece é a existência de uma “parede” que separa aulas teóricas de aulas práticas na computação. Por esse motivo, os alunos ficam desmotivados e acabam por optar pela desistência do curso, pois a parte teórica é difícil e sem a motivação de saber onde usar toda aquela teoria, ela se torna um peso sem sentido. Achar formas de unir essas disciplinas, praticar essa matemática em programação, mostrar a real utilidade, se torna fundamental para maior engajamento dos alunos nessas aulas e consequentemente nos cursos.



Matemática no dia a dia do desenvolvedor

Saindo do meio acadêmico, chegamos no meio profissional. E com isso, a grande questão: A matemática influencia no dia a dia de um desenvolvedor? Ao meu olhar, sim! Existem alguns casos onde a área de desenvolvimento não a utiliza, e isso é aceitável. Porém o número de áreas que a utilizam é bem maior. Por exemplo, não é necessário saber calcular uma integral por partes “na mão” para programar uma camada de serviços de API. Porém o exercício mental e lógico que é feito para calcular a integral, a forma de tentar achar uma solução para resolvê-la pode ser o grande diferencial na programação. Além disso, seu tempo para achar uma maneira otimizada para fazer o código pode ser bem menor caso você já tenha resolvido problemas matemáticos como exercício. Isso se chama lógica matemática e é a base de desenvolvimento de softwares.

Aplicações reais da matemática na computação

A lógica matemática é somente uma das reais aplicações na computação. A mais forte talvez seja os ramos da Inteligência Artificial. Não tem como trabalhar com códigos de IA sem ter um bom embasamento de matemática, sendo fundamental saber cálculo, álgebra linear e probabilidade. Em assuntos de IA, o nível de matemática é tão alto que é comum que matemáticos ou estatísticos por formação migrem para essa área, pois é necessário pessoas que saibam lidar com toda essa teoria.

Outro exemplo de aplicação é de quem cria códigos de interface, tanto em web ou mobile. Este desenvolvedor se vê muitas vezes fazendo cálculos matemáticos para posicionamento de elementos na tela e responsividade. Em jogos, a matemática aparece em grande escala como o uso da álgebra

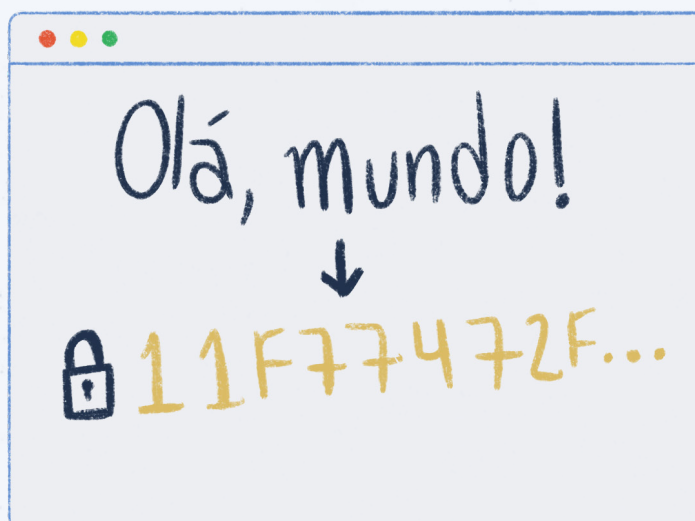


```
INICIO
  inteiro x,y,i,r
  ler x
  ler y
  para i de 1 até x
    se (x%i=0) e (y%i=0) então
      r ← i
    fim se
  proximo
  escrever r
fim
```

$O(n!)$

$\Omega(n^2)$

$\Theta(2^n)$



linear para definição de elementos na tela. É necessário saber noções sobre matrizes porque a tela nada mais é do que uma grande matriz, onde pixels são os valores de cada célula. Além disso, os movimentos dos jogadores também são calculados. A rotação de elementos visuais necessita de conhecimentos de trigonometria para saber a taxa exata de rotação.

Na Segurança da Informação, temos a criptografia como forma essencial de criar sequências numéricas que não podem ser decifradas. Logo, além da teoria dos números, são usados conceitos de números primos. É usado o fato de ser quase impossível fatorar um número em primos. Com isso, a criptografia usa dessa característica para criar números gigantescos (na escala de milhões de algarismos) e usá-los como chaves para transações bancárias, por exemplo. Este é um tema super vasto e interessante, o qual recomendo maiores estudos por parte do leitor.

Em computação aplicada temos a de Aná-

lise de algoritmos, uma área fundamental no desenvolvimento, que nos mostra a qualidade do nosso código, se ele é eficiente ou não. São usados conceitos de funções matemáticas, pois os níveis são distribuídos por funções, como por exemplo a linear ou exponencial.

Conclusão

Por fim, a grande lição é que a computação é uma aplicação matemática. Logo, precisamos dela como nossa aliada. Saber apreciar a beleza da matemática é o primeiro passo. Depois é usá-la a seu favor. Esses passos podem ser um diferencial no seu dia a dia como desenvolvedor. Para fechar, uma frase de Alan Turing: “Raciocínio matemático pode ser considerado um tanto esquematicamente como o exercício de uma combinação de duas instalações, o que podemos chamar de intuição e criatividade”. Bom, o que seria um desenvolvedor sem intuição e criatividade? 🧐

Um mapa da *UX research*

TEXTO *Carolina, Elis, Lucas Alves*

EDIÇÃO *Amanda*

ILUSTRAÇÃO *Carolina*



Imagine que seu trabalho no momento é projetar a melhor experiência possível para um aplicativo destinado à adestradores de cachorros. Qual o primeira coisa a fazer? Um protótipo de baixa fidelidade? Definição de escopo? Bem, provavelmente vai ser difícil fazer qualquer coisa antes de entender a respeito das necessidades dessa profissão e como um aplicativo pode resolvê-las. E você só tem como apreender isto com pesquisas de usuário.

O exemplo acima é uma situação dentre muitas que nós designers nos encontramos quando recebemos um projeto, e em situações como essa, precisamos coletar informações com o público-alvo objetivando conseguir projetar algo que traga a melhor experiência possível para os usuários. As informações coletadas serão necessárias durante todo o processo de desenvolvimento, desde a estratégia até os últimos detalhes da interface visual. Por esse motivo, a realização de pesquisas bem feitas e bem elaboradas se tornam tão importantes.

É muito comum, principalmente com profissionais de Design que estão no início dos estudos, dar pouca importância às pesquisas realizadas com usuários, que são, ou deveriam

ser, o cerne das atividades de todo UX Designer. Uma fala dada por um dos alunos do Academy IFCE exemplifica isso, “Mesmo quando finalmente aprendo a respeito, e incluo algum tipo de pesquisa em nosso processo, na maioria das vezes essas pesquisas se resumem a testes de interface e usabilidade.”, disse Ronaldo Gomes. E apesar de importantes, estes tipos de pesquisa não são as únicas quando falamos sobre UX Research.

UX Research ou pesquisas de experiência do usuário é uma grande área dentro do processo de experiência do usuário que consiste na realização de investigações sobre o público-alvo para a coleta de informações e assim estruturar ideias baseado no seu modo de pensar, nas suas necessidades e em como eles interagem com sua interface, e assim, conseguir projetar uma solução eficiente. Mas, como escolher a pesquisa mais adequada para projeto em questão?

O tipo de pesquisa a escolher depende dos objetivos que se deseja alcançar, pois algumas técnicas podem ser encaixar melhor com os resultados esperados, como pesquisas de mercado para a construção da estratégia, ou

card sorting para ajudar na arquitetura de informação. Mas isso não são regras, por isso é importante conhecer bem as metodologias para conseguirmos aplicá-las quando necessário, e focar no objetivo em mente.

Em geral, as pesquisas podem ser de natureza quantitativa com geração de dados numéricos e assim poder quantificar o problema ou qualitativa onde é feito uma pesquisa exploratória por meio de razões, opiniões, motivações. Estas pesquisas quantitativas e qualitativas podem ouvir o que os usuários tem a dizer sobre o que pensam (attitudinal) ou somente observar o que eles fazem, uma pesquisa comportamental (behavioural).

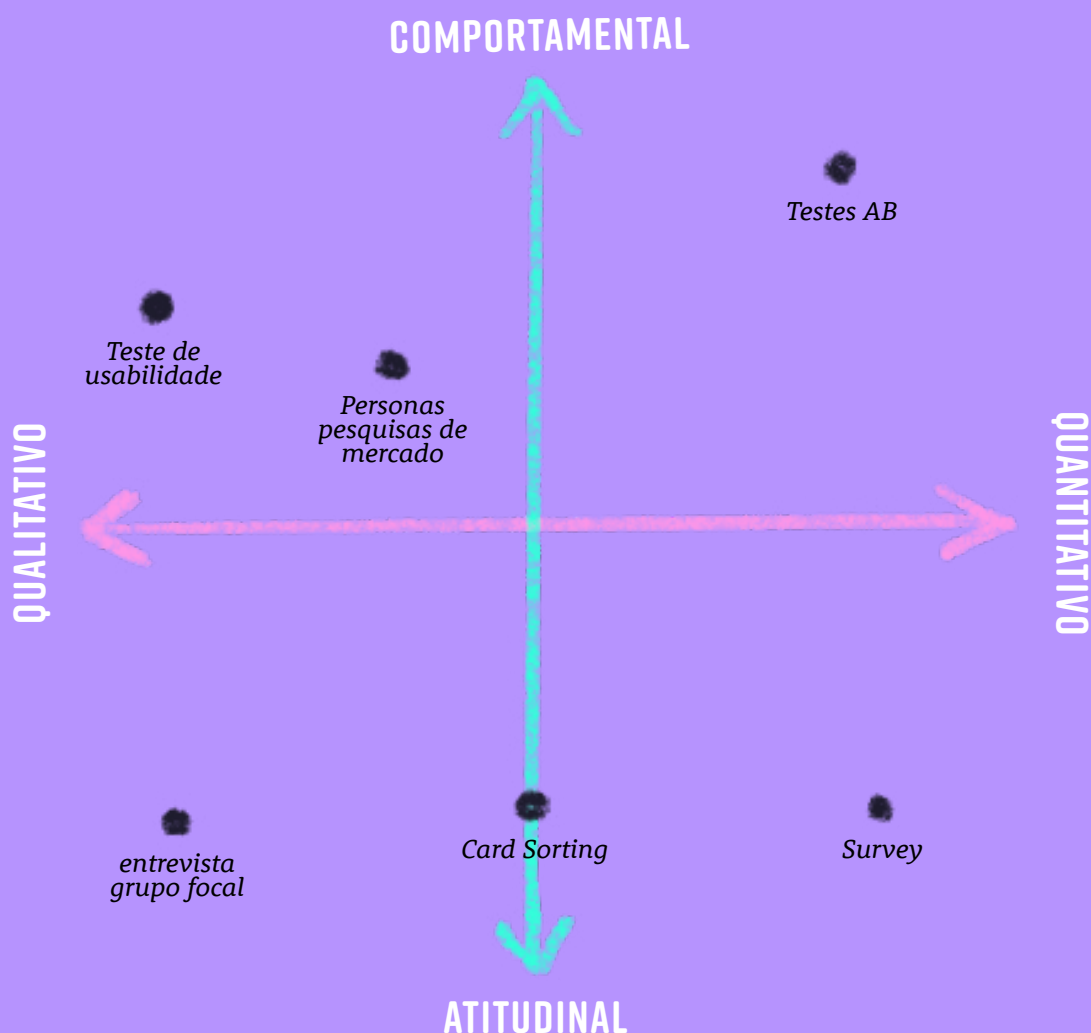
A pesquisa de UX é uma pesquisa como qualquer outra. É importante aplicar todos os cuidados éticos para com os participantes e dar a devida importância aos momentos de preparação dos instrumentos, formulação das perguntas e análise dos dados. Um certo rigor técnico só irá contribuir para a validade da pesquisa e será importante para que se obtenha resultados, de fato, relevantes.

Algumas coisas para se pensar:

1) Se for gravar, sempre pedir permissão para os participantes. E no início da pesquisa, sempre apresentar como os dados serão utilizados. Nunca pergunte mais do que você irá precisar. Por que perguntar o CPF da pessoa, ou até mesmo o nome, se isso for irrelevante para a pesquisa?

2) Prepare com antecedência os instrumentos e ambiente da pesquisa. Vai utilizar algum software, fez um roteiro de perguntas? Teste tudo isso antes. Testar seus instrumentos faz com que você encontre erros antes que estes erros afetem o resultado da sua pesquisa.

3) E falando em perguntas, é muito importante que elas não direcionem o usuário à uma resposta específica. Os participantes de uma pesquisa costumam, mesmo sem querer, responder o que eles acreditam que o entrevistador quer ouvir. Não pergunte coisas como “Você gostou da experiência deste aplicativo?”.





Pesquisas de Mercado

Técnica usada para pesquisar sistemas semelhantes no mercado, com o objetivo de coletar informações como parâmetro de análise e até mesmo descobrir oportunidades de soluções ou melhorias no design do sistema que vai ser colocado ou já está no mercado.

Ferramentas: excel (construção de tabelas - inventário de recursos, matriz de análise comparativa), miro.



Personas

Personas são uma representação de dados reais coletados a partir de um contato direto ou indireto com o usuário. Exibindo mais claramente dados demográficos bem como as necessidades do usuário. Facilitando a adaptação do projeto ao público-alvo.

Ferramentas: diagrama de afinidades, smaply, Make My Persona.



Entrevistas

Técnica de coleta de dados e levantamento de requisitos, onde o avaliador irá guiar uma conversa com o usuário por meio de perguntas, sendo estas abertas ou fechadas. Uma entrevista pode ser estruturada ou semi-estruturada.

Ferramentas: bloco de notas com perguntas previamente definidas



Surveys

Questionário com perguntas fechadas, normalmente de natureza quantitativa. Muito utilizado para explorar problemas e pensamentos do público-alvo, descobertos anteriormente por meio de pesquisas qualitativas. Normalmente os surveys são preenchidos digitalmente.

Ferramentas: Surveymonkeys, Google forms, Typeform.



Card Sorting

Exercício rápido e de baixo custo com o objetivo de entender como o usuário estrutura ou organiza um grupo de itens ou até mesmo rotula um conteúdo, para ajudar a gerar ideias e soluções que aumentem a capacidade de localização do usuário dentro de um sistema.

Ferramentas: optimalworkshop, usabilitytest, userzoom, miro, figma, post-its



Grupos Focais

Reunião com um grupo formado idealmente por 5-10 usuários, um moderador, um facilitador e um pesquisador. A conversa deve ser documentada e guiada pelos seus organizadores, permitindo a coleta de dados diversos pelos pesquisadores.

Ferramentas: bloco de notas com perguntas previamente definidas.



Teste de Usabilidade

Observação do usuário por um ou mais avaliadores no momento de uso do sistema que está sendo avaliado. Este teste tem como foco avaliar a usabilidade de um sistema interativo a partir das experiências de uso dos usuários, podendo ocorrer de forma local ou remota.

Ferramentas: lookback, loom, mazedesign, usabilityhub.



Testes AB

Observação do usuário no momento de uso do sistema que está sendo avaliado, onde são testados dois caminhos de navegação (com o mesmo objetivo) dentro do sistema e o mais usado pelo usuário será analisado pelo avaliador(es).

Ferramentas: optimizely, AB testy.

Sonda Cultural: A Nascente dos Rios

TEXTO *Livia Oliveira*
EDIÇÃO *Helaine*
ILUSTRAÇÃO *David*

A maior perspectiva do design é mapear o sentimento das pessoas através de perguntas que alcançam as suas experiências e as estendem em pesquisa. A exploração do território vem da busca em entender comportamentos que revelam vivências, histórias e contextos dos participantes em questão. Escolher métodos de pesquisa engajados permitem novas formas de experiência, diálogo e consciência sobre a problemática a ser imergida. Isso nos propõe a ultrapassar todas as soluções prefiguradas que nos tendenciam ao projetar, para que seja possível construir junto com as pessoas intervenções em problemas que ainda virão a ser evidenciados. A nossa tarefa é florescer práticas experimentais e colaborativas que engajem emocionalmente comunidades a compartilhar sobre seu cotidiano com o intuito de co-criar sob seu território.

Diante disso, os cinco meses de relações inteiramente remotas devido ao COVID-19, me fez querer entender a viabilidade em realizar pesquisas de campo e testes não-moderados à distância com usuários. E, para além, também fazer com que a pesquisa de campo potencializasse os usuários a se tornarem participantes dos processos. Como moradora de uma cidade litorânea com essência estuarina, onde a transição entre o rio e o mar marca a vegetação e

a vida urbana, tenho as águas como paixão. A partir desse campo que me situo, decidi estruturar uma investigação que engajassem a comunidade, ao passo que são construídos dados sobre como as pessoas se relacionam com os Rios no Recife. A ideia é que esses participantes fossem unidos por um eixo territorial, nesse caso a Região Metropolitana, e tivessem propósitos semelhantes; vivenciar, cuidar ou projetar para os Rios da cidade.

Para me inserir nesse contexto, escolhi a Sonda Cultural como ferramenta co-criativa, ela funciona como uma pesquisa qualitativa semi-etnográfica para produzir dados centrados nos colaboradores de forma individual e mais autônoma possível. Esse método tem como foco a participação das pessoas na autodocumentação, busca descrever fenômenos humanos através do cotidiano e das interações, como também incentiva a explorar novas possibilidades para problemas até então desconhecidas. Essa maneira de levar em conta as experiências e emoções dos participantes foi fundamental para querer testar se é possível resgatar remotamente.

Misturar métodos presenciais que exigem entrevistas e materiais físicos como caneta, papel e câmera para adequá-los a realidade remota se tornou o grande desafio. Não perder a essência do projeto em construir relações profundas e



afetivas com os participantes em seus territórios, sem que seja preciso sair de casa, exigiu que fluxograma fosse posto em prática. O encontro entre cinco bacias constrói a essência da cidade do Recife. Em busca desse encontro que meu aprendizado começou. Vamos navegar juntos nessas descobertas?

Nascente dos Rios

Onde as ideias chegam na superfície.

Em busca de navegar sobre as diferentes realidades que vivenciam diariamente os Rios Capibaribe, Beberibe, Tejiptó da cidade do Recife, me cerquei de perguntas para investigar e agregar sentidos às etapas que ainda não estavam definidas. Essas perguntas se iniciaram em saber quais motivações existiam para deixar de aplicar uma pesquisa quantitativa tradicional, pautadas em longas entrevistas com profundidade. Depois me questionei como faria para aplicar um material remoto que os participantes respondessem ao longo do seu dia, de forma livre, sem a minha interferência como pesquisadora. Isso tendo como base o interesse e o envolvimento pelo tema que pauta a vida dessas pessoas para me provocar a escutar com atenção, afeto e, assim, querer projetar com elas.

Esse Rio que nasce em Poção, olho d'água do interior de Pernambuco e margeia todos os bairros do Recife, consegue unir diversos atores que se comunicam no cotidiano da cidade através das águas. Essas pessoas, por vezes, cresceram fazendo do assoalho marinho a subsistência de vida em suas comunidades ribeirinhas; utilizam os rios como meio de transporte; projetam ações urbanas inseridas nas águas ou olham todos os dias para essa paisagem natural da sua janela. Pensar nisso também me fez questionar se eu conseguiria alcançar remotamente diferentes contextos sociais, econômicos e políticos. E como eu poderia sintetizar os materiais diversos compartilhados pelos participantes para mostrar as relações intrínsecas das suas rotinas, memórias, registros e encontros que convergem e divergem entre si.

Simbora viver a prática! Preparados para se inserirem no contexto e sondar territórios com as pessoas?

O projeto Nesse Recife Tem Rios foi criado para materializar a sonda no contexto que ela deverá ser inserida, nesse caso, se conectar com pessoas que possuem diferentes relações com os Rios da cidade. Entregar as perguntas na mão dos participantes é a forma mais profunda e real

“Rio Capibaribe é a história de uma vida. As vias fluviais me conectaram com o Recife. Foi o momento onde conheci a cidade por outra perspectiva.” Davi, barqueiro no Jardim do Baobá.

de entender suas relações nos espaços. Como a maioria dos participantes não me conheciam, antes de enviar a sonda os contactei por mensagem ou ligação, dependendo da faixa etária e familiaridade com as tecnologias, para saber se tinham disponibilidade em participar dessa troca comigo. Já que estavam me conhecendo pela primeira vez, também compartilhei vivências que aproximam os nossos propósitos de acordo as diferentes realidades.

Depois de ambos os lados quererem se conhecer, a sonda se inicia com um convite individual para situar as pessoas no ambiente confortável que gostaria que elas se sentissem ao partilhar comigo suas vivências. Isso ocorre sem interferências de quem construiu o material, nesse caso, a forma como os participantes respondem as perguntas cabem a eles e o tempo que devolveriam a sonda também. Queremos imergir nas suas rotinas para aprender através das experiências, então esse material afetivo perpassa por suas imagens, localidades, opiniões, memórias, distância e percurso. Questões essas que considero relevantes para evidenciar inúmeras realidades. Ao longo do dia, senti como se tivesse sido recebida em suas casas para tomar um café, ouvindo o que cada um dia para partilhar.

Afluentes

Quem se une para formar os Rios da cidade.

É nesse momento que os atores se unem como corpo que sustenta o fluxo das águas. Aplicar a Sonda Cultural me levou ao encontro com pessoas que constroem todos os dias suas relações com os Rios. A primeira conversa foi Luciana, protetora dos manguezais e defensora de um Recife Sem Lixo, que mobiliza desde familiares a instituições sociais para retirar uma tonelada de resíduos na margem do rio Capibaribe. Em

seguida, Geisiane, dona do Bistrô NegraLinda, a marisqueira mais conhecida do Recife, responsável por reverter através do empreendimento social a violência que durante anos fez parte da Ilha de Deus.

Encontrei também com Mila, arquiteta e urbanista, que (re)projetou o Recife como uma Membrana Anfíbia capaz de sobreviver a elevação do nível das águas e aquecimento global. Como não bastasse, escutei Duda, criadora do bloco flutuante A Ver Capibaribe e incentivadora de passeios piratas pelo Rio. Ela também me apresentou Davi, barqueiro no Baobá, que considera o Capibaribe a história de uma vida e celebra casamentos enquanto navega. E se eu dissesse que Adriana também os conhece? Ela é educadora e projeta transportes públicos navegáveis. Bem, se além disso tudo, eu tivesse a sorte de Mayra, acordaria todos os dias tomando café na beira do Rio e cuidando dos resíduos da comunidade do Monteiro.

Cada uma dessas histórias são capazes de revelar os emaranhados que as unem e as tornam dissonantes nas suas relações com o território, reconheci principais fatores que marcam a essência do que me foi sondado qualitativamente. A dependência para tirar o sustento dos rios e sobreviver a partir dele; inspiração para projetar ou tomar café da manhã sob as margens; vulnerabilidade em ter menor abastecimento das águas ou sofrer devastações nas moradias devido as cheias acíclicas. Esses fatores confluem no sentimento que envolvem todos a viverem os rios a sua maneira.

Margem das Águas

Lugar que o Rio encontra os seus contextos.

As margens e o leito situam o lugar que o Rio ocupa em seu maior encontro. Para conseguir acessar as relações existentes nesse lugar, foi necessário se permitir flutuar na memória e mergulhar nas vivências das pessoas. Sem registros fotográficos, áudios, textos, desenhos e mapas, não conseguiria criar laços tão profundos capazes de reconhecer o contexto de encontros que o Rio as coloca.



Esse vídeo do Nesse Recife Tem Rios é um compilado das imagens compartilhadas pelos participantes que definem seu envolvimento com as águas.



Adriana, educadora!
Casa Amarela - 2.2km

Mila, inspirada!
Setúbal - 3km



Duda, navegadora!
Parnamirim - 214m



Mayra, apaixonada!
Monteiro - 0km



Geisiane, cozinheira!
Ilha de Deus - 0km



Luciana, protetora!
Iputinga - 2.2km



Davi, barqueiro!
Ilha de Deus - 0km



O que isso nos mostra?

Encontro com a foz dos rios.

É possível acessar remoto muitas realidades. As diferentes informações que conseguimos receber agregam e fazem com que projetemos melhor com e para as pessoas. Utilizar-se de metodologias mais participativas e integrativas, como a Sonda Cultural, são práticas que empresas como a Google e Pagseguro já aderiram. Essa flexibilidade permite que os processos de campo sejam mais ágeis, sem longas entrevistas, mas não percam a profundidade e o envolvimento fundamental para se construir relações. Segundo Tim Ingold, antropólogo em seu diálogo com design, afirma que isso reflete na observação participante do pesquisador, ir além da coleta de dados e se permitir aprender ao passo somos educados pelas pessoas. Podemos ser a foz dos rios, união das águas doces e salgadas, para desaguar no oceano atlântico. Se propor a conectar histórias, informações e ideias que culminam no grande aprendizado que são as pessoas.

Construir diferentes métodos para aplicar e testar em ambientes que fomentam inovação na tecnologia foi a possibilidade que a **@AppleDeveloperAcademy|UFPE** me deu. Convido vocês também a serem designers experimentais e utilizarem processos participativos para se permitirem novos diálogos, experiências e consciência sobre o território imergido.

Agradeço aos que fizeram parte de todo o processo, me fizeram passear pelos seus Rios e também inspiram de diversas formas a ouvir e projetar com eles. ❶

Nesse Recife Tem Rios

Mae West

"Keep a diary, and someday it'll keep you."

Mantenha um diário e algum dia ele o manterá.



Simple Diário

Crie e mantenha seus registros diários
de uma maneira simples, rápida e customizável.

Escaneie para download na App Store



Protegido com Biometria



Vários temas customizáveis



Registro de humor e clima



Design Patterns: o que são e quais os benefícios

TEXTO *Patrícia*
EDIÇÃO *Beatriz*
ILUSTRAÇÃO *José João*

Os tão famosos padrões de projeto ou design patterns são um conjunto de ideias que surgiram para ajudar a resolver problemas comuns no design de software. Essas ideias já foram amplamente testadas, melhoradas e agora estão prontas para serem implementadas deixando seu código fácil de entender e reutilizar.

O arquiteto Christopher Alexander foi quem apresentou a ideia de patterns em 1970 para construir um vocabulário comum para discussões sobre design. Ele escreveu: “Cada pattern descreve um problema que ocorre várias vezes ao nosso redor e, com isso, descrevem a solução para o problema de uma maneira que você pode usar essa solução diversas vezes sem ter que fazer a mesma coisa duas ou mais vezes”.

Então, na verdade, podemos aplicar os patterns para todas as categorias de problemas. Porém, tenho uma ressalva, você precisa conhecer seu problema, generalizar e então aplicar um padrão apropriado. Afinal, nem todos os problemas precisam de um padrão de solução.

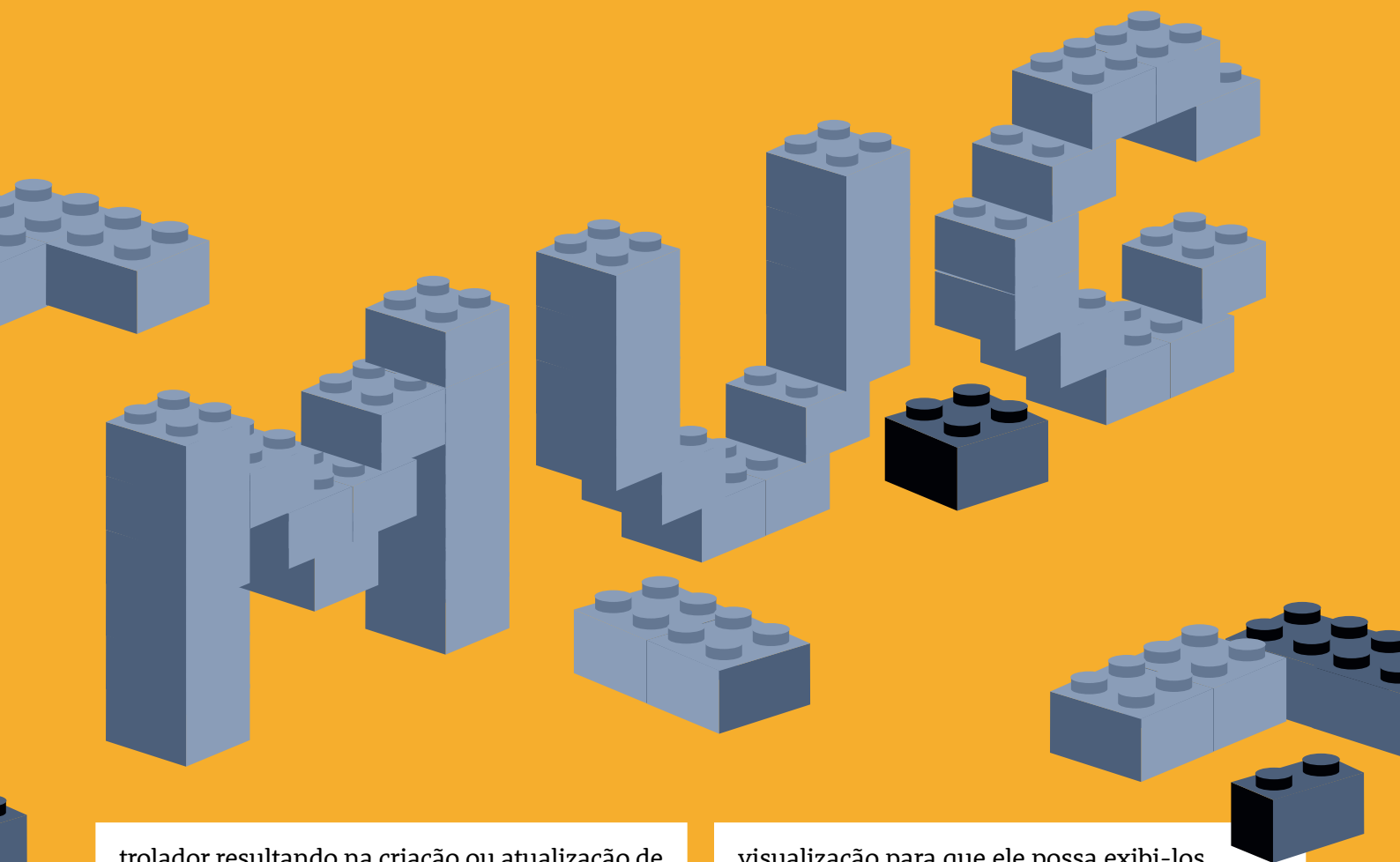
O objetivo desse artigo é olhar de perto os três principais padrões arquiteturais utilizados em aplicações iOS e fazer um breve comparativo entre eles.

Cocoa MVC (Model-View-Controller)

Dentre a monarquia do mundo do desenvolvimento, Model-View-Controller (MVC da Apple) é o rei dos patterns, sendo um dos blocos de construção do Cocoa e uma variação do tradicional MVC. Ele estabelece não apenas regras para o funcionamento de objetos na aplicação, mas também o melhor caminho de comunicação entre eles. Há inúmeros benefícios de adotar esse padrão, por exemplo, os objetos tendem a ser mais reutilizáveis e suas interfaces melhor definidas. Vamos passar por cada camada explicando o conceito proposto por esse padrão de projeto.

Os modelos encapsulam dados específicos de uma aplicação. Idealmente, um modelo não deve ter conexão explícita com os objetos de visualização que apresentam seus dados, portanto, ele não deve se preocupar com problemas de interface e apresentação do usuário.

As ações do usuário acontecem na camada de visualização, onde são criados ou modificados os dados e são comunicadas através de um con-



trolador resultando na criação ou atualização de um modelo. Então quando um modelo muda, ele notifica um controlador, que atualiza os objetos de visualização apropriados.

Um objeto de visualização é um objeto que sabe como desenhar a si mesmo e pode responder às ações do usuário. Um dos principais objetivos da camada de visualização é exibir dados dos objetos modelo do aplicativo e permitir a edição desses dados.

Os objetos de visualização aprendem sobre alterações nos dados do modelo através dos controladores do aplicativo e comunicam alterações iniciadas pelo usuário também através do controlador para os objetos do modelo de um aplicativo.

Um controlador atua como intermediário entre um ou mais objetos de visualização de uma aplicação e um ou mais modelos. Ele também podem executar tarefas de configuração e coordenação para um aplicativo e gerenciar os ciclos de vida de outros objetos. Basicamente, ele interpreta as ações do usuário feitas em objetos de visualização e comunica alterações. Quando os objetos do modelo mudam, um controlador comunica esses novos dados do modelo aos objetos de

visualização para que ele possa exibi-los.

Como as visualizações não referenciam modelos diretamente, a tarefa de formatação e apresentação costuma ser delegada ao controlador, assim como operações de networking, gerenciamento de eventos, etc. Esta organização acaba se tornando um problema, pois ainda que o desenvolvedor tenha a opção de quebrar sua interface no que são conhecidas como ‘Subviews’, o que é visto na prática são os famosos controladores massivos, fazendo com que os desenvolvedores busquem alternativas ao padrão MVC na comunidade ou em outros padrões arquiteturais já consagrados.

Acrescenta-se ao fato o quão difícil será testar responsabilidades de componentes quando estes possuem muitas responsabilidades, o que é o caso do controlador, demandando atenção por parte do desenvolvedor, que deve escrever testes não só para validar modelos, como também para testar funcionalidades e a fidelidade visual para diferentes tipos de telas.

Contudo, o MVC continua sendo a uma boa opção para projetos pequenos que não exigem arquiteturas mais elaboradas e que não têm a intenção de se preocupar com um processo de manutenção rigoroso.

MVVM

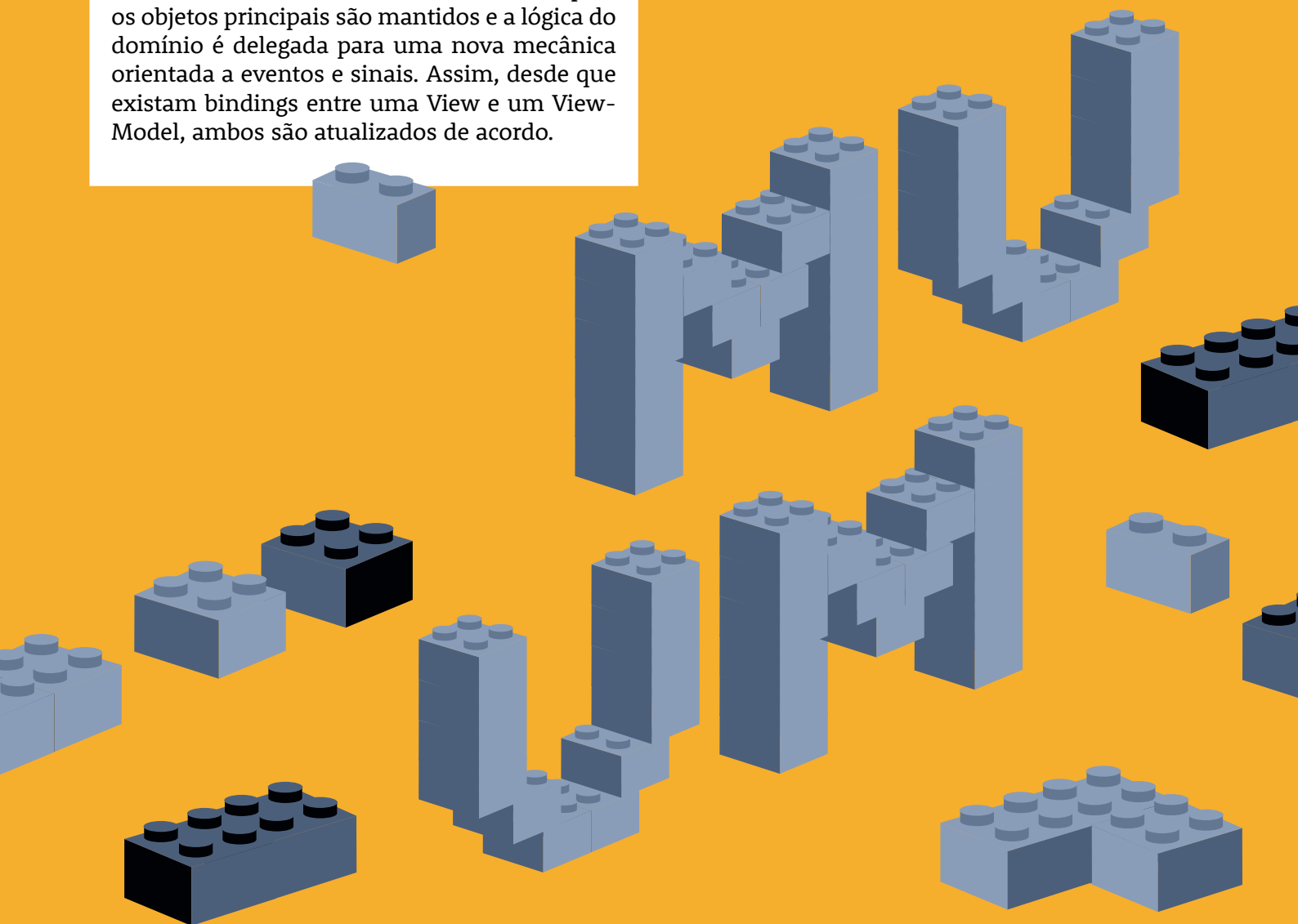
(Model-View-ViewModel)

A grande jogada do padrão de arquitetura MVVM se dá na proposta de diminuir o código excessivo de apresentação das visualizações, fazendo com que o componente ViewModel se encarregue pela apresentação. Nesta variação MV*, iniciou-se o conceito de ViewModel, um mediador entre visualizações e modelos que substitui os controladores apresentados no padrão MVC.

A facilidade de manutenção do código é evidente sendo possível entrar em partes menores e especializadas do código e fazer alterações por causa da separação de tipos de códigos de maneira limpa. Assim será possível lançar novas versões com facilidade.

Em algumas situações, o MVVM pode ser visto como uma forma evoluída do Cocoa MVC, pois os objetos principais são mantidos e a lógica do domínio é delegada para uma nova mecânica orientada a eventos e sinais. Assim, desde que existam bindings entre uma View e um ViewModel, ambos são atualizados de acordo.

“CADA PATTERN DESCREVE UM PROBLEMA QUE OCORRE VÁRIAS VEZES AO NOSSO REDOR E, COM ISSO, DESCREVEM A SOLUÇÃO PARA O PROBLEMA DE UMA MANEIRA QUE VOCÊ PODE USAR ESSA SOLUÇÃO DIVERSAS VEZES”



Viper

(View, Interactor, Presenter, Entity e Routing)

O Viper veio com dois grandes objetivos: resolver o problema dos controladores massivos e melhorar a testabilidade da aplicação. As cinco camadas são:

View: Exibe o que é ordenado pelo Presenter.

Interactor: Contém regras especificadas para um caso de uso.

Presenter: Contém a lógica de visualização e prepara elementos para serem exibidos baseando-se em dados obtidos do componente Interactor.

Entity: Contém modelos básicos usados no Interactor.

Routing: Contém a lógica de navegação e a ordem de exibição de telas.

Este padrão arquitetural é bastante popular pelo fato de aproximar o desenvolvedor dos conceitos apresentados pela arquitetura limpa e que por sua vez é fundamentada em princípios de design orientado a objeto.

A introdução de Entities e Routers é a grande jogada desse padrão, assim como a possibilidade da separação de responsabilidades em compo-

nentes tão pequenos quanto se queira. O componente Interactor cuida da lógica relacionada às entidades, networking ou persistência. Com isso, faz com que as entidades no modelo Viper sejam implementadas como modelos.

A grande diferença entre Viper e as arquiteturas MV* está na responsabilidade destinada ao Presenter. Apesar de ser independente do UIKit, o Presenter possui a responsabilidade de gerenciar lógicas relacionadas à UI. Já as mecânicas que favorecem navegação e roteamento, o componente responsável é o Router.

Dentre suas desvantagens temos a grande quantidade de código que sempre é repetido sem nenhuma alteração em qualquer cenário que utilize tal arquitetura. Também possui um número excessivo de interfaces para classes com responsabilidades pequenas. Importante citar o componente Presenter, que possui fluxos de dados multidirecionais, o que diverge do proposto na Arquitetura Limpa, que encoraja o uso de fluxos de controle unidirecionais.

Avaliando as características do Viper podemos verificar que este modelo arquitetural pode ser considerado excessivo para aplicações simples.

Conclusão

Uma das maiores dificuldades dos desenvolvedores está em criar código que evolua bem e resista ao tempo à medida que a complexidade dos seus sistemas aumenta. Ao escolher um padrão arquitetural, o desenvolvedor visa criar código que não seja hostil, de modo que as manutenções e a adição de novas funcionalidades se torne um trabalho mais fácil. Segregar componentes de acordo com suas funcionalidades é um processo vital à saúde do software. ❶

Quem é o mais queridinho?

TEXTO *José João, Helaine e Felipe*

EDIÇÃO *Mateus Rodrigues*

ILLUSTRAÇÃO *José João*

Quem é o melhor... Xibs, Storyboard ou View Code? Esta é uma discussão sem fim em todos os Academies. Por isso perguntamos a 93 alunos e alunas de Academies do Brasil tudo para saber suas opiniões: Afinal, qual é o mais queridinho?

MAS DIZ AÍ, QUAL É O SEU
QUERIDINHO?

VIEW CODE
86%

XIBS

STORYBOARD

VOCÊ ACHA QUE É MAIS USADA
NO MERCADO?

VIEW CODE
73.3%

XIBS

STORYBOARD





The Universe

Descubra informações e curiosidades
sobre o universo, além de ver
fantásticas imagens reais.

Seja beta! baixe o app no Testflight e
nos dê Feedback sobre sua experiência,
desta forma poderemos entregar um app
cada vez melhor para você!





**BAIXE AQUI OS NOSSOS WALLPAPERS COM AS
ILUSTRAÇÕES DESTA EDIÇÃO DA SANDBOX!**

