

2021

**SARA
LARDANI**

Educação Ambiental Digitalizada.

Processo de implementação de uma aplicação didática para sensibilização de crianças.

2021

**SARA
LARDANI**

Educação Ambiental Digitalizada.

Processo de implementação de uma aplicação
didática para sensibilização de crianças.

Dissertação apresentada ao IADE - Faculdade de Design, Tecnologia e Comunicação da Universidade Europeia, para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Design de Interação realizada sob a orientação científica do Doutor José Braga de Vasconcelos, *Professor Auxiliar, Faculdade de Design, Tecnologia e Comunicação (IADE) da Universidade Europeia.*

Dedico este trabalho a minha família.

agradecimentos

Começo por agradecer aos meus pais e a minha família que me ensinaram a nunca desistir.

Agradeço a Francesco que sempre esteve perto de mim em todos os momentos, principalmente nos momentos em que me sentia derrotada por tudo, inclusive por mim mesma.

Agradeço as minhas amigas maravilhosas por nunca duvidarem de mim.

Agradeço a todas as novas amizades nascidas neste novo caminho de vida por terem contribuído para o meu crescimento pessoal.

Por último, agradeço ao meu orientador por toda a disponibilidade demonstrada.

-

Ringrazio i miei genitori e la mia famiglia che mi hanno insegnato a non mollare mai.

Ringrazio Francesco che mi è stato sempre accanto in ogni momento, soprattutto in quei momenti in cui mi sentivo sconfitta da tutto, compresa me stessa.

Ringrazio le mie meravigliose amiche per non aver mai dubitato di me.

Ringrazio tutte le nuove amicizie nate in questo nuovo percorso di vita per aver contribuito alla mia crescita personale.

Infine ringrazio il mio relatore per tutta la disponibilità dimostrata.

palavras-chave

Educação Ambiental; Processo de Design; Tecnologias de informação; Linguagem de programação *Processing*

resumo

Atualmente, o ambiente requer enorme atenção.

Depois de mais de um século de poluição, produção intensiva, pesca e consumo ilimitado de plásticos de uso único, estamos a destruir nosso planeta que precisa de mudanças decisivas de comportamento da humanidade.

Iniciar o processo de sensibilização da sociedade desde o ensino primário para tornar as crianças mais conscientes do problema, é uma estratégia relevante. A evolução tecnológica e o advento da internet levaram a uma mudança do ambiente escolar, convertendo métodos tradicionais de ensino, permitindo o acesso a recursos e serviços de aprendizagem fora da estrutura escolar. É, portanto, apropriado analisar a integração das Tecnologias de informação e comunicação (TIC) nos processos de educação, tanto do ponto de vista prático como pedagógico. As crianças tendem a ser fascinadas pela última geração de tablets e dispositivos móveis e, desde a sua tenra idade, começam a formar preferências e competências e a expressar as suas emoções através da palavra.

Esta tese tenta descrever, da forma mais completa possível, o processo de criação de aplicativos para crianças que sejam eficazes, mas também divertidos. Será dada particular importância aos princípios do Design de Interação e às teorias pedagógicas para garantir que a experiência vivida com a ferramenta de ensino atenda às necessidades e competências da criança.

Finalmente, considerando necessário o entendimento básico do designer da linguagem de programação Java Script, analisaremos algumas interações básicas deste ponto de vista.

Keywords

Environmental education; Design Process; Information technologies; Processing programming language

abstract

Currently, the environment requires huge caution. After more than one century of pollution, intensive production, fishing, and unlimited use of disposable plastic, we are destroying our planet, which needs immediate changes in human behavior. Developing an awareness process in society, starting from primary school, is a remarkable strategy.

Technology developments and the Internet have brought a significant change in the educational system, disrupting traditional methods and allowing external resources and services within educational facilities. Therefore, it is desirable to assess ICT solutions (Information and Communication Technologies) in educational processes, both from a pragmatical and a pedagogical viewpoint.

Kids tend to be attracted by last-generation tablets and smartphones, and, from an early age, they start having preferences and abilities to express their emotions through words.

This work tries to describe, in a comprehensive way, the creation process of apps for kids, both effective and entertaining.

Interaction Design principles and pedagogical theories will be emphasized to ensure that the experience with the educational tool meets the kid's requirements and expectations.

In conclusion, considering that a basic knowledge of JavaScript programming language is needed, we will analyze a few interactions from this point of view.

Índice

1. Introdução

iii

1.1 Motivação para o estudo

iii

1.2 Objetivos da dissertação

iii

1.3 Organização dos capítulos

iv

2. A crise Ambiental e a Educação Ambiental

v

2.1 A problemática ambiental

v

2.2 O movimento Greta Thunberg

viii

2.3 Tecnologia de informação na educação da crianças

ix

3. Processo de criação de uma aplicação didática

xii

3.1 UX Child Center Design para educação ambiental

xii

3.2 Metodologias de Design Process

xvii

3.3 Análise da concorrência

xxii

3.4 Arquitetura do sistema

xxv

4. Processo de Design

xxvii

4.1 Wireframe	xxviii
4.2 Linguagem visual	xxix
4.3 Criação de interface gráfica	xxxiv
4.4 Unmoderate test	xxxvi
4.5 Moderate test	xxxvii
5. Prototipagem	xxxix
5.1 Ambiente software Processing	xlii
5.2 Biblioteca P5.js	xliv
5.3 Processo de prototipagem com P5	xlvi
6. Conclusões	lxiv
7. Bibliografia	lxvii

1. Introdução

1.1 Motivação para o estudo

Em 1969, William B. Stapp, pela primeira vez, introduziu o termo educação ambiental e, a partir desse momento, iniciou um processo de mudança na maneira de ver e considerar a educação ambiental. Foi reconhecida como uma ferramenta fundamental para a aquisição de consciência e conscientização sobre questões ambientais e tornou-se um meio para desencadear uma mudança de comportamento em relação ao ambiente.

Acredita-se que uma das tarefas de quem estuda design de interação é melhorar a vida humana, eu quero melhorar a vida humana melhorando as pessoas. Aumentar a consciencialização ambiental das crianças deve ser o primeiro passo para alcançar um mundo melhor. A promoção de comportamentos ambientais são necessários para o bem do futuro deste planeta.

Existe uma necessidade real de fazer algo pelo nosso planeta. Uma população crescente, produção em massa, o uso de recursos naturais, e produção de resíduos, nomeadamente plásticos descartáveis e políticos descartáveis perturbaram o equilíbrio do mundo. Além disso, os políticos demoraram muito tempo a estarem conscientes do problema.

Se continuarmos a queimar combustíveis fósseis, destruir florestas, produzir alimentos de forma intensiva e comer de forma insustentável, destruiremos nosso futuro. Para uma política mais verde, além de políticos sensibilizados é necessário que também os cidadãos mudem os seus comportamentos. A idéia é consciencializar as crianças por meio de educação ambiental, a fim de criar futuros cidadãos que estejam cientes da situação e de como se comportar da maneira correta.

1.2 Objetivos da dissertação

O presente trabalho de tese trata da identificação de uma solução digital para o ensino da educação ambiental nas escolas.

As crianças estão na base do futuro da nossa sociedade e é necessário dar às crianças todos os meios para poderem aprender, conhecer e estudar todos os aspectos do nosso mundo. A

tecnologia é o meio que nos permite fazer isso sem comprometer a qualidade e a quantidade de informações.

Nesta tese queremos analisar como criar uma ferramenta educacional adequada para uso por crianças, analisando cada fase do processo de design.

Vamos, portanto, analisar quais são as metodologias fundamentais para realizar este processo de criação de ferramentas, avaliando os melhores comportamentos a serem seguidos ao projetar aplicações para crianças.

Na verdade, tanto do ponto de vista da linguagem visual, como do ponto de vista das interações criança-app a serem utilizadas, como finalmente do ponto de vista dos métodos de avaliação de usabilidade, é necessário seguir algumas regras para proceder de maneira Child-Friendly, ou seja proceder de maneira a que as crianças se sintam confortáveis.

1.3 Organização dos capítulos

A tese foi escrita com uma estrutura linear com o objetivo de mostrar as fases fundamentais do processo de criação de uma aplicação didática para crianças.

O primeiro capítulo será dedicado ao principal problema identificado para o nosso estudo: a crise ambiental e a necessidade de sensibilizar as crianças.

Analisaremos aqui quais são as necessidades sociais e educacionais com um olhar particular na situação italiana antes e depois da pandemia de Covid-19.

No segundo capítulo mostra-se a primeira fase do processo de criação de uma aplicação educacional, que é a análise do mercado e das necessidades do nosso target, identificando duas metodologias distintas de processos de design.

No terceiro capítulo descreve-se a parte mais criativa do processo de design, mostrando as várias componentes do design na quais é necessário prestar mais atenção e os testes de usabilidade necessários para entender melhor se o que estamos a criar está de acordo com o uso pelo nosso alvo.

Por fim, no quarto capítulo mostra-se a fase de prototipagem por meio de programas reais de programação utilizando a linguagem Java (Processing) e a linguagem JavaScript (P5.js).

Esses programas foram criados especificamente para aproximar o mundo dos designers do mundo dos programadores. Neste capítulo, mostra-se a história da prototipagem e como passamos a usar esses programas ao longo dos anos.

Também mostra-se como usar o programa p5.js para aproximar as crianças ao mundo da educação ambiental por meio da criação de interações simples, mostrando também as strings de código necessárias.

Durante a apresentação das várias fases do processo de Design, usaremos muitas imagens como exemplos dos vários procedimentos

2. A crise Ambiental e a Educação Ambiental

2.1 A problemática ambiental

Existe uma necessidade real de fazer algo pelo nosso planeta.

O desafio ambiental, ligado à conservação dos recursos do nosso planeta, representa um desafio que não pode ser evitado, se queremos respeitar as gerações futuras.

Os problemas ambientais que o nosso planeta enfrenta são muitos. Eles ameaçam seriamente o ecossistema, ou seja, todas as formas de vida que vivem num delicado equilíbrio. Percebeu-se que os recursos da Terra não são ilimitados e que não podem ser explorados indefinidamente. Além disso, a consciência de que todos os ambientes naturais do planeta estão interligados e que quando o equilíbrio de um ecossistema é afetado, isso determina consequências em cadeia que afetam outros ecossistemas.

Estamos em uma era que impõe no mundo inteiro escolhas radicalmente diferentes das que foram feitas no passado: longe do modelo de produção tradicional, voltado para um novo modelo de economia que respeita o ambiente, orientado para uma sociedade que não produz resíduos, mas sabe como criar riqueza e bem-estar com a reutilização e reciclagem de materiais e com a

regeneração de recursos. Para que isso aconteça, é necessária uma profunda mudança de mentalidade que envolva instituições, empresas e indivíduos. E essa nova consciencilização ambiental deve começar com escolas e estudantes de todas as idades. Sobretudo pelos mais jovens, o que poderíamos chamar de "nativos ambientais": uma geração que na rotina diária de comportamento já encontra respeito pelo ambiente em que vive como uma perspectiva natural.

Internacionalmente, essas preocupações foram reiteradas nas inúmeras conferências realizadas, resultando em importantes resoluções, que no entanto, nem sempre foram cumpridas, como: a Convenção sobre Mudanças Climáticas, a Convenção sobre Diversidade Biológica, a Declaração do Rio, a Declaração das Florestas e a Agenda 21. Estes instrumentos de política internacional têm, no entanto, uma contribuição relevante para a identificação de problemas e desenvolvimento de uma consciência ambiental cada vez mais madura.

A definição de educação ambiental deve-se a William B. Stapp, da Universidade de Michigan:

"A educação ambiental tem como objetivo produzir uma cidadania consciente e interessada no ambiente e em seus problemas, consciente em como ajudar a resolver esses problemas trabalhar para a solução deles".

O Ministério do Ambiente e a Proteção do Território e do Mar (italiano) define a educação ambiental:

“uma ferramenta fundamental para conscientizar os cidadãos de maior responsabilidade pelos problemas ambientais e conscientizar sobre a necessidade de se envolverem nas políticas ambientais do governo territorial. A educação ambiental não é um estudo simples do ambiente natural, mas deve promover mudanças nas atitudes e comportamentos individuais e coletivos”.

Na União Europeia, a educação ambiental tornou-se parte integrante das atividades curriculares das escolas primárias e secundárias: muitos Estados-Membros introduziram a educação ambiental nas suas escolas como uma oferta de treino didático. O estudo do ambiente visa

preparar os alunos para a construção de um futuro verde e a promoção de uma vida numa sociedade mais sustentável. Internacionalmente, a estratégia da UNECE (Comissão Económica das Nações Unidas para a Europa) para a educação para o desenvolvimento sustentável (Vilnius, 2005), definiu a educação ambiental como um pré-requisito para o desenvolvimento sustentável ou como uma ferramenta para a boa governação e para processos de tomada de decisão.

A educação ambiental como tal nasceu na década de setenta do século passado. Desde então, muitas maneiras de a praticar foram experimentadas por diferentes atores e em diferentes contextos. Em cerca de trinta anos, houve uma enorme proliferação de experiências. Os atores multiplicaram-se ao longo do tempo: escolas de todos os níveis, terceiro setor com associações ambientais e Organizações Não Governamentais de Ambiente (ONGAs), alguns movimentos políticos, autoridades locais etc.

A educação ambiental (EA) é interpretada como uma ferramenta para promover um comportamento que respeite os ambientes naturais e artificiais. A sua missão é disseminar práticas virtuosas para o ambiente. Isso aconteceu durante quase toda a década de 1980. No entanto, o risco de tal abordagem é cair na demagogia, ou permanecer superficialmente ancorado nas regras a serem cumpridas, sem se preocupar muito que haja uma compreensão amadurecida dos problemas da base. Assenta numa tendência de querer dirigir os outros, de querer "manipular" as consciências. Esta é a principal limitação desta abordagem. No entanto, esta abordagem pode ser melhorada e superada por meio de práticas e metodologias de reflexão compartilhada, baseadas na empatia e na experiência.

O modelo dialógico é o modelo que se consolidou na década de 90 e que atualmente é considerado uma referência para a maioria dos atores da área de EA. É uma aplicação estrutural do modelo de educação ecológica com particular atenção aos aspectos éticos e sociais. O tema assume um papel central e a reflexão educacional está profundamente entrelaçada com a ambiental no sentido de impulsionar mudanças comportamentais, tanto a nível individual como social. A ênfase agora é na forma como os processos educacionais são conduzidos, na sua

qualidade. Não importa tanto a quantidade de informações transmitidas, mas acima de tudo, a qualidade do processo de aprendizagem.

Este é o modelo que orienta atualmente a reflexão sobre EA não só a níveis nacionais, mas também internacionais.

2.2 O movimento Greta Thunberg

O planeta e a natureza são o nosso bem mais precioso que compartilhamos com todas as criaturas vivas, mas nós, os humanos, estamos há anos destruindo-os. Talvez a determinação da jovem sueca Greta Thunberg tivesse que chegar para despertar as consciências e nos lembrar o sentido de pertencermos à Terra e a responsabilidade que temos de cuidar mais dela. FridaysForFuture é um movimento que começou em agosto de 2018. Foi iniciado por Greta Thunberg, uma rapariga de 16 anos que decidiu sentar-se diante do parlamento sueco por três semanas, para protestar contra a falta de políticas reais sobre a questão climática. Ela postou o que estava fazendo no Instagram e no Twitter o seu protesto tornou-se viral. O mundo definitivamente a notou quando ela disse no Fórum de Davos "vocês estão a roubar o futuro de vossos filhos". Greta fez escola. Não só as hashtags #FridaysForFuture e #Climatestrike se espalharam como um incêndio, mas jovens de diversos países, e depois também adultos, começaram a protestar em frente aos parlamentos e municípios de seus países.

Seu apelo foi claro e direto e não mediu as palavras: “o planeta está morrendo e os governos devem agir”. O seu claro apelo à participação, e a fazer parte de um processo regenerador do mundo, também se fez sentir entre os adolescentes e pré-adolescentes que se manifestaram espontaneamente em várias manifestações por todo o mundo, expressão de uma geração atenta, sensível e respeitosa. Hoje fala-se menos sobre Greta e sua batalha, nos jornais e nas notícias, mas o processo de consciencialização sobre as questões ambientais já começou.

E é justamente na consciência que precisamos nos concentrar, na necessidade de criar um pensamento intrínseco nas pessoas. Educar não para ensinar conceitos e dados, mas para criar um hábito de consciencialização.

É justamente na educação que nos precisamos de focar para que seja normal que as novas gerações façam a reciclagem, fechem a torneira enquanto tomam banho, andem mais a pé e de bicicleta em vez de ir de carro.

Os hábitos, bem o sabemos, são difíceis de combater, por isso o mais inteligente a se fazer é acostumar as crianças a se comportarem de uma determinada maneira, para que possam viver neste mundo sem o destruir. O objetivo deve ser fazer com que eles protejam a terra sem esforço e quase sem perceberem que o estão a fazer. Agir espontaneamente é provavelmente a maneira mais eficaz. Fazer o bem a nós mesmos e ao mundo sem nenhum esforço, sem a ideia de que é um dever, comportarmo-nos com responsabilidade sem sentir o peso da responsabilidade.

A questão é: a tecnologia e a digitalização podem ajudar nesse objetivo?

2.3 Tecnologia de informação na educação da crianças

Enriquecer a experiência de ensino com as TIC é uma necessidade inevitável numa ideia de escola que pretenda enfrentar e gerir os desafios de uma sociedade cada vez mais moldada pelos ritmos e regras da rede.

Os nativos digitais desenvolvem formas de identidade individual e social muito diferentes das dos adultos. Vamos observar como as crianças "usam" as telas sensíveis ao toque: sua "alteridade antropológica" (Ferri, 2011) é agora um fato evidente. Basta observar que as crianças, na Itália e no mundo, já desenvolvem um relacionamento privilegiado com telas interativas desde 2003 (Ferri, Mantovani, 2006, 2007, Ferri, 2014).

Em relação a isso, a escola está mudando tem mudado muito ao longo dos últimos anos.

Em toda a Europa, as aulas são abolidas, os espaços abertos e as oficinas para trabalhos em pequenos grupos nascem, o ensino encontra novas variações nas aulas virtuais. O espaço físico

da aprendizagem e a tipologia dos objetos culturais que compõem o mix de ensino são redefinidos nas novas oportunidades oferecidas pela tecnologia: Internet, iPod, quadros digitais e e-learning, juntamente com as lições e livros frontais.

A relação cada vez mais precoce e estruturante entre crianças e telas sensíveis ao toque determina novas emergências também em relação aos estilos de aprendizagem. De fato, novos estilos cognitivos peculiares e específicos estão a surgir - e relevantes nesse sentido são as contribuições de Katie Davies e Howard Gardner (Davies, Gardner, 2013). Esta nova situação representa um desafio e uma oportunidade também para a pesquisa no campo educacional. De fato, as práticas de comunicação e interação on-line mediadas por telas interativas parecem harmonizar-se bem com uma abordagem pedagógica que encontra sua matriz original em um certo ativismo típico da aprendizagem do fazer (Dewey, 1938) e na epistemologia do sócio-construtivismo contemporâneo (Bruner, 1990; Brown, Campione, 1994; Gardner, 1983).

A partir do século XVII, e até à Revolução Industrial, a invenção da imprensa causou uma mudança no paradigma escolar, focando-se em ferramentas didáticas como o livro e a aula (frutos pedagógicos da modernidade). Agora o digital está a transformar radicalmente a educação, trazendo para o centro ferramentas como a Internet, plataformas de comunicação / compartilhamento e redes sociais. Não porque a tecnologia orienta a educação, mas porque a educação, imersa neste tempo, valoriza a tecnologia, valorizando-se a si mesma.

A escola italiana anterior ao COVID-19 resistia fortemente às mudanças, empoleirada em sua tarefa tradicional de alfabetização e, portanto, inclinada a se defender dos perigos do digital, escalando os bastiões da disciplina e do suposto dever de transmissão cultural.(R.Franchini 2020)

Promover a disseminação do ensino invertido no mundo era justamente a possibilidade, que hoje se tornou uma necessidade premente com o surgimento do Coronavírus, de acessar com flexibilidade conteúdos online, disponíveis a qualquer hora e de qualquer dispositivo graças à disseminação de plataformas digitais como Coursera, Edx, Udacity, Khan Academy, OpenUpEd, EduOpen ou o italiano WeSchool que oferecem a possibilidade de seguir ou criar,

em alguns casos, aulas e cursos universitários e não universitários. A usabilidade e acessibilidade destes conteúdos a todos foi a pré-condição para podermos imaginar novas metodologias de ensino e que, como afirma Salman Khan, fundador da Khan Academy, ao permitir que os professores tivessem mais tempo para apoiar os alunos permitiu aos próprios professores "*usar a tecnologia para tornar as salas de aula mais humanas*". (E. Sabatini, 2020)

O Ensino à Distância através das tecnologias digitais, que o sistema escolar nacional teve que adotar com a maior urgência após o bloqueio total de todas as atividades formativas presenciais e o isolamento preventivo necessário para conter a dramática disseminação do COVID-19, tem ativado debates, comparações, reflexões muito estimulantes sobre o potencial que essas tecnologias podem ter num futuro não muito distante (espera-se) pós-pandemia.

A pergunta que se impõe é se este novo método funcionou de forma eficiente, nomeadamente na Itália? Não, parece ser a resposta.

Seguindo as diretrizes ministeriais, os professores continuaram a ensinar de uma forma que os colocava no centro do ensino, adaptando a tecnologia aos seus propósitos e métodos usuais. Os professores não questionaram como transformar a educação graças à tecnologia, não explorando o potencial desta que pode ser aproveitado para imaginar novas situações de formação que antes eram inconcebíveis.

A tecnologia, em vez disso, poderia ter mudado o foco do ensino, centrando-se no aluno, que se deve sentir parte do processo e poderia estar mais aberto à discussão e à aprendizagem graças à curiosidade que se instalou nele.

Vários estudos e relatórios destacaram as oportunidades e os benefícios potenciais das tecnologias de informação e comunicação (TIC) para melhorar a qualidade da educação. As TIC são vistas como uma "ferramenta principal para a construção de uma sociedade do conhecimento" (UNESCO 2003, 1) e, em particular, como um mecanismo ao nível da escola que pode proporcionar uma forma de repensar e redesenhar sistemas e processos educativos, conduzindo à qualidade. educação para todos.

A tecnologia pode até representar um terceiro professor, que é um recurso adicional na sala de aula, capaz de apoiar e ajudar as crianças e jovens nos estudos, integrando no projeto educacional e formativo de uma pessoa o que antes era representado pelo espaço simples e pelos materiais usados na escola, como afirma Loris Malaguzzi, professor e pedagogo. Também pensamos apenas em quão útil um computador ou tablet pode ser para uma criança com dificuldades de aprendizagem específicas (SLD), quando o software pode transformar um texto longo numa síntese que se pode ouvir. Também, em vez de um discurso frontal do professor, pode criar-se um mapa conceptual ou infográfico, e essa é uma imagem evocativa que se pode partilhar com os alunos que também podem usá-la em casa para estudar.

As novas tecnologias na sala de aula permitem realizar simulações, viajar e orientar-se, encontrar informações de diferentes fontes e compará-las, escrever textos de forma cooperativa, assistir a video tutoriais e realizar exercícios interativos, ou seja, permitem-nos experimentar tarefas autênticas e dinâmicas, experiências que exigem que os alunos se envolvam ativamente usando ferramentas que lhes são familiares.

3. Processo de criação de uma aplicação didática

3.1 UX Child Center Design para educação ambiental

A tecnologia usada no estudo de algumas disciplinas é, portanto, capaz de integrar a experiência de aprendizagem escolar e oferecer um ponto de partida para chegar a um conhecimento significativo.

O uso de ferramentas tecnológicas de educação pode, portanto, ser um excelente método de ensino de educação ambiental, utilizando-as em paralelo com experiências ativas na natureza. A experiência interativa durante o aprendizagem pode ser um método para criar aquela consciência intrínseca considerada essencial para estabelecer uma consciência ambiental na pessoa.

Mas como criar um instrumento adequado para crianças entre 10 e 12 anos?

O período entre 10 e 12 anos é uma fase particular da idade das crianças, principalmente se pensarmos em sua relação com a tecnologia. Frequentemente, não há informações para

aplicativos para crianças dessa idade, pois tendemos a concentrar-nos nas faixas etárias em que as crianças são mais novas e os pais e educadores seguem cada vez mais de perto as opções de aplicativos e ferramentas tecnológicas. À medida que as crianças crescem, elas também se tornam mais autônomas na sua abordagem ao mundo da tecnologia e dos aplicativos móveis, mas isso não significa que não haja possibilidade de criar um produto adequado às suas necessidades.

A principal diferença entre o design para crianças e o design para adultos está nos objetivos dos utilizadores. De acordo com Debra Gelman, autora de *Designing for Kids: Digital Products for Playing and Learning* (Gelman 2014), ao projetar para adultos, mesmo ao projetar jogos para adultos, o objetivo é ajudá-los a cruzar a linha de chegada. Mas quando se projeta para crianças, a linha de chegada é apenas uma pequena parte da história.

De acordo com Gelman, existem 4 diferenças principais a serem consideradas ao projetar um aplicativo para crianças:

- Crianças adoram desafios.

Na verdade, de acordo com Gelman, os microconflitos ajudam as crianças a resolver seus conflitos internos e ajudam as crianças a desenvolver habilidades como:

- Prever como os outros provavelmente reagirão ao seu comportamento
- Controle de suas emoções
- Comunique-se claramente
- Ver a opinião de outras pessoas
- Resolver desacordos de forma criativa

- Crianças querem feedback

Ao brincar num espaço digital, as crianças esperam um feedback visual e auditivo sempre que fazem algo. É por isso que a maioria dos aplicativos infantis gera algum tipo de reação ou resposta a cada interação. As crianças esperam ser recompensadas pelo que fazem.

- As crianças confiam mais do que os adultos. Como as crianças não podem prever ou compreender as consequências de suas ações com antecedência, geralmente são muito mais confiantes do que os adultos.

- Crianças aprendem mais rápido.

Uma boa regra prática: concentre-se em uma faixa etária de 2 anos, no máximo. Pense nas diferenças entre uma criança de 4 e 8 anos. Enquanto uma faixa etária pode mergulhar e aprender o aplicativo à medida que avança, outra pode precisar de instruções claras para aumentar sua confiança ao usar o aplicativo.

No entanto, também existem muitas semelhanças com o design para adultos. Aqui também Gelman esclarece as 4 principais semelhanças entre o design para crianças e para adultos:

- Os utilizadores esperam consistência

De acordo com Gelman, crianças e adultos incomodam-se com elementos de design que parecem aleatórios e desnecessários. As crianças gostam de objetos na tela para fazer coisas interessantes. "Elementos que atrapalham ou ganham vida espontaneamente ou que não contribuem para o objetivo geral podem frustrar crianças e adultos e induzir o abandono do jogo ou aplicativo. Além disso, se tudo na tela se mover, ficar colorido ou fizer barulho no mesmo nível, crianças e adultos ficarão confusos sobre o que é interativo e o que não é, e isso torna muito difícil para que usem o site ou aplicativo. Um princípio de design comum para adultos é manter as interações e os comentários consistentes para que os utilizadores possam aprender rapidamente como usar o site ou aplicativo. O mesmo vale para as crianças."

- Os utilizadores precisam de um motivo para usar o aplicativo

As crianças ficarão entediadas rapidamente se não tiverem um motivo para usar um aplicativo.

- Os utilizadores não querem ser surpreendidos

Os utilizadores imaginarão que o seu aplicativo funciona de determinada maneira e esperam que funcione dessa forma. Eles não se querem surpreender quando ele se comporta de maneira diferente do que esperavam. O mesmo vale para as crianças, diz Gelman, e dá o seguinte exemplo: “Quando uma criança adiciona joias a uma caixa num jogo, espera poder abrir a caixa onde as joias estão armazenadas para as ver todas tirar as gavetas e procurar as coisas que pensa que lá estão”.

- Os utilizadores querem algo mais

Gelman chama isso de *lagniappe*. Um *lagniappe* é algo mais, um ovo de Páscoa, lançado para encantar utilizadores ou clientes, e adultos e crianças aproveitam essas pequenas interações inesperadas que aprimoram sua experiência com um site ou aplicativo.

"Projetar um bom aplicativo para crianças é mais do que apenas silenciar conceitos adultos".

Cit. Debra Gelman

Para projetar para adultos, é preciso entender os utilizadores e onde eles estão cognitivamente, fisicamente e emocionalmente. No entanto, as crianças podem ser difíceis de compreender, especialmente se não as vivenciarmos pessoalmente, por isso é necessário observá-las especialmente para compreender os seus hábitos.

Criar personas de design de nossos utilizadores ideais será necessário para os entender e reconhecer suas necessidades, encontrar necessidades latentes e detectar suas expectativas.

Personas são perfis inventados que descrevem os diferentes tipos de utilizador de um serviço, relatando suas motivações, expectativas, necessidades e atitudes. As *personas* geralmente nascem a partir de resultados de pesquisas e observações de campo, agrupando perfis de utilizadores com características semelhantes. Cada agrupamento é descrito pela criação de um único personagem exemplar, que destaca os traços de atitude (necessidades, expectativas, problemas, etc.) compartilhados por todas as pessoas pertencentes a esse grupo.

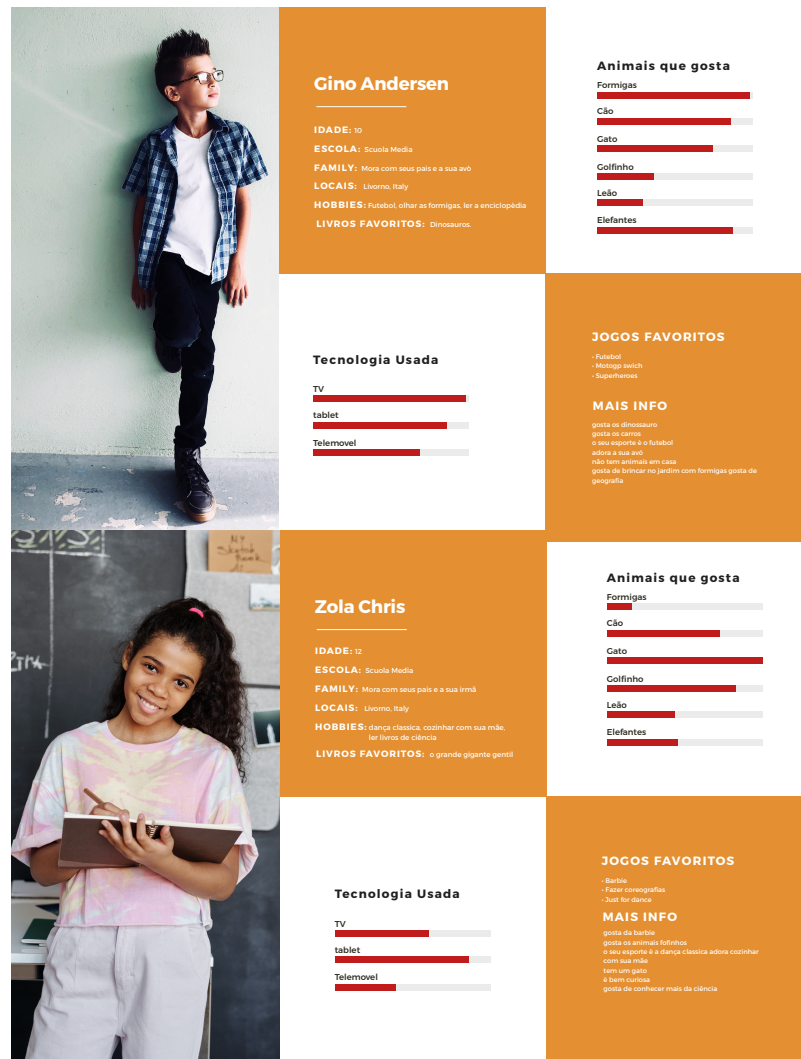


Fig. 1 Exemplos de personas criadas durante o projeto desenvolvido durante o mestrado em design de interação

As *personas* ajudam a gerar ideias a partir das necessidades existentes: podem ser usadas para mergulhar no lugar do utilizador durante um brainstorming ou atividade de workshop e criar novos recursos e serviços que atendam às suas necessidades e expectativas. (.it designers italia)

3.2 Metodologias de Design Process

Design Thinking, termo cunhado em 1990 por David Kelley e Tim Brown (Kelley, Brown, 1990), é uma abordagem que estuda e analisa problemas e intervém nos processos que os afetam, eliminando obstáculos e encontrando novas oportunidades de melhoria. Este método explora as dimensões estratégicas nas quais o projeto se enquadra e usa imaginação, criatividade, intuição e lógica com análise e planeamento. As principais características do design thinking são: foco na análise de problemas, deixando de lado a ideia de uma solução e envolver todas as pessoas que estão diretamente interessadas no que se está a planear, para ver o tema de diferentes ângulos.

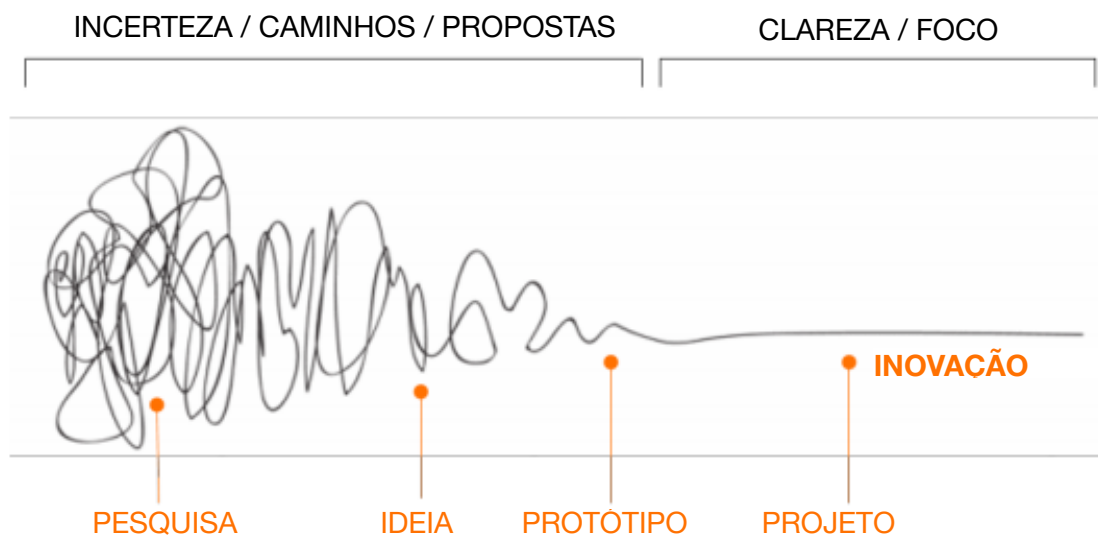


Fig. 2 Da incerteza à inovação (_____)

A solução, também chamada de Big ideia, é apenas a última etapa do processo que primeiro tenta analisar o tipo de problema e quais são os nós que impedem a solução. A identificação destes nós permite encontrar soluções, verificar a sua validade e trazer inovação. O Design Thinking traz uma solução totalmente baseada nas necessidades do utilizador final. É ele quem está no centro do processo.

É uma metodologia muito poderosa porque usa a palavra "empatia" em seu verdadeiro sentido.

- **Empatia:** nesta primeira fase de análise, a situação é explorada, imersa no contexto de referência e tentando compreender totalmente o ambiente, as pessoas e os contextos em que as soluções devem ser encontradas.
- **Definir:** Todas as informações encontradas na fase de empatia são reunidas para definir os principais problemas identificados até ao momento. Esta fase ajuda a reunir ideias para estabelecer recursos e funções que permitem solucionar problemas com o mínimo de dificuldade.
- **Conceber:** depois de compreender o utilizador e as suas necessidades e analisar os problemas, pode proceder à concretização de ideias, identificar novas soluções e experimentar novas alternativas à visão dos problemas. Nenhuma solução é descartada e as ideias que surgem e são mais interessantes são submetidas a análises para verificar a sua viabilidade e vantagem competitiva.
- **Protótipar:** as ideias ganham forma com a construção de protótipos, versões de baixo custo que permitem trabalhar as soluções para o problema criado na fase anterior. Esses protótipos podem ser testados entre membros do grupo, por outros departamentos ou por pequenos grupos de pessoas fora da equipe de design. Esta etapa tem por objetivo identificar a melhor solução possível para cada problema identificado acima. Cada solução é implementada, revisada, aceita ou melhorada com base na experiência do utilizador. Desta forma, tem-se uma visão melhor de como as pessoas podem se comportar interagindo com o produto que está a ser criado.
- **Teste:** uma vez que as ideias tenham sido materialmente realizadas, elas devem ser testadas, possivelmente melhoradas e testadas novamente, se necessário. Esta fase também serve para entender o utilizador, as condições de uso, como as pessoas pensam e como se comportam.

O processo de Design Thinking parece delineado de forma linear, onde uma fase conseqüentemente leva à próxima até a conclusão. Na prática, as coisas não são bem assim, na verdade o processo pode ser realizado de forma mais flexível, por exemplo, desenvolvendo várias fases ao mesmo tempo ou coletando informações ainda durante as fases de prototipagem.

O design thinking é um processo iterativo e não linear. Isso significa simplesmente que a equipe de design usa continuamente as suas descobertas para rever, questionar e melhorar as suas suposições, entendimentos e resultados iniciais.

Os resultados da fase final do processo de trabalho inicial informam a compreensão do problema, ajudam a determinar os parâmetros do problema, permitem redefinir o problema e, talvez o mais importante, fornecem novos insights para que se possam ver soluções alternativas que poderiam não estar disponíveis com o nível de compreensão anterior.

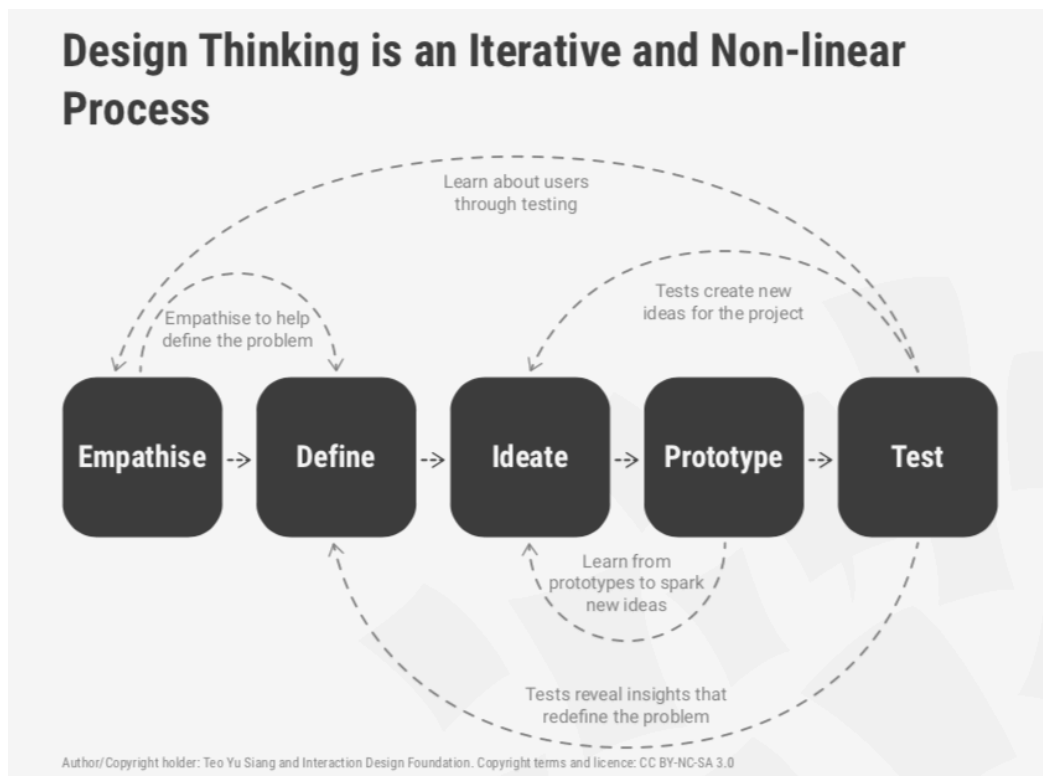


Fig 3. Design Thinking (The Basics of User Experience Design By INTERACTION DESIGN FOUNDATION 2020)

Outra metodologia muito poderosa é a desenvolvida por Tiago Franco, Beatriz Costa e Maria Grilo no livro *Product Design Process: The manual for Digital Product Design and Product Management* (2019).

Um manual de design de produto digital baseado em uma abordagem multidisciplinar que une proprietários de produtos, designers, desenvolvedores e gerentes. Tem como foco a excepcional experiência do utilizador e o design de interação, mas também na manutenção do equilíbrio entre custos e benefícios, evitando desperdício de recursos e apostando no rápido lançamento no mercado. Uma abordagem que nasce da experiência e do uso de metodologias como *Design thinking* e UX.

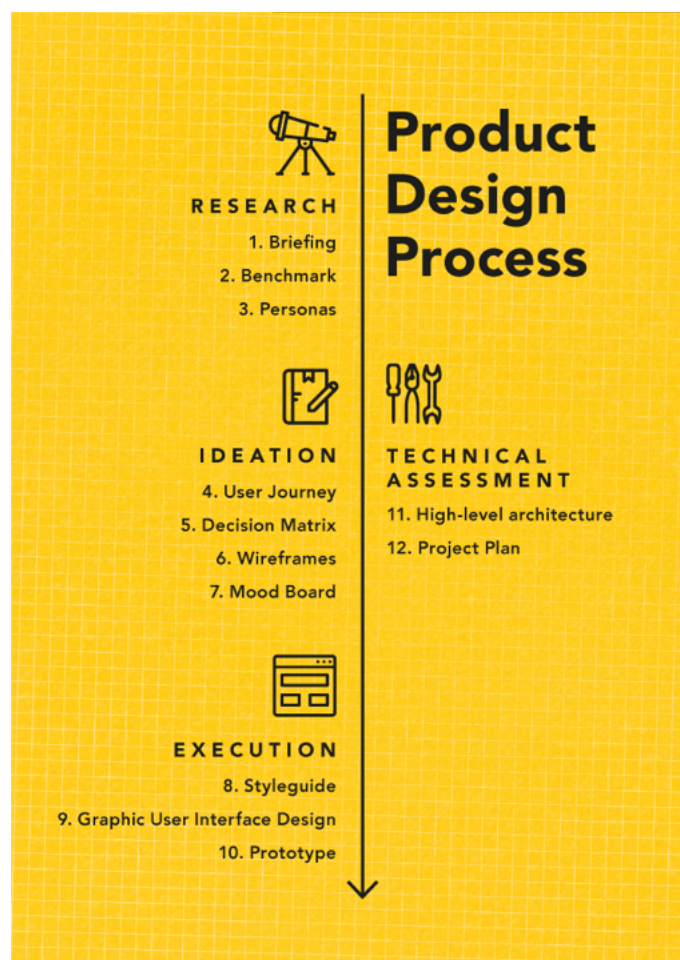


Fig. 4 Product Design Process : The manual for Digital Product Design and Product Management

Após inúmeros projetos, clientes e stakeholders acreditam ter chegado à escolha das fases de desenvolvimento mais eficientes e eficazes possíveis: com base nas necessidades do consumidor, colaboração constante com o cliente e as várias figuras profissionais em ação e elaboração constante de relatórios.

Sua abordagem é baseada em várias etapas práticas não lineares:

- **Pesquisa**

A primeira fase do PDP é de responsabilidade do designer UX e seu objetivo é reunir evidências que irão apoiar as decisões tomadas a partir de agora, garantindo que nenhuma decisão seja tomada com base em suposições vagas. É a fase em que são identificados os principais aspectos do modelo de negócio e as necessidades do utilizador.

1. BRIEFING
2. USER RESEARCH
3. DESIGN BENCHMARK

- **Idealização**

A ideação é o fulcro do processo criativo e é aqui que o conceito do produto será formulado com base nas necessidades do utilizador e do modelo de negócio, ambos identificados na fase de pesquisa. Aqui, o designer UX, o designer visual e o proprietário do produto devem trabalhar juntos.

4. USER JOURNEY
5. DECISION MATRIX
6. WIREFRAMES
7. MOOD BOARD

- EXECUÇÃO

Esta fase é da responsabilidade do designer visual e tem como foco a criação de uma representação física do conceito que foi definido até aqui.

8. GUIA DE ESTILO

9. DESIGN GRÁFICO DA INTERFACE DO UTILIZADOR

10. PROTÓTIPO

- Avaliação técnica

O objetivo desta fase é garantir que todos os requisitos e ideias gerados sejam realistas em sua implementação e viáveis, considerando o tempo e o orçamento disponíveis.

11. HIGH-LEVEL ARCHITECTURE

12. PROJECT PLAN

A Avaliação Técnica deve ser realizada simultaneamente com as 3 primeiras fases e é muito importante ter uma Equipe Técnica envolvida na Equipe de Projeto.

Frequentemente, os designers não percebem o que uma ideia simples envolve em termos de orçamento, planeamento e tempo.

3.3 Análise da concorrência

Quando se decide desenvolver um aplicativo, existem várias etapas a serem seguidas para um design adequado, evitando assim perda de tempo, dinheiro e recursos. A análise da concorrência potencial e a análise SWOT são duas etapas fundamentais em qualquer estratégia de marketing para coletar informações competitivas, descobrir os pontos fortes e fracos dos concorrentes, a fim de entender como entrar no mercado.

Uma cuidadosa análise competitiva permite identificar eventuais obstáculos e aprofundar o conhecimento do mercado, o que é útil para a tomada de decisões precisas e lucrativas. A análise da concorrência pode ser efectuada directamente navegando nas lojas e estudando aplicações

semelhantes ao que está a ser desenhado, verificando a oferta existente e o comportamento do consumidor.

BENCHMARK			
	Animal 360	Wenno	Baby Zoo
Info	✓	✓	✓
Video	×	×	×
Qualidade	×	×	×
Importância Ecológica	×	×	×



Fig. 5 Exemplo Benchmark criado durante o projeto desenvolvido durante o mestrado em design de interação

Um sistema útil de comparação das várias aplicações presentes é a criação de uma tabela que lista todos os possíveis concorrentes com as suas características, permitindo assim evidenciar os pontos fortes e fracos, as oportunidades e riscos (análise SWOT), a sua presença e difusão. No mercado, o público-alvo e comentários deixados pelos utilizadores A análise SWOT é uma ferramenta de planeamento estratégico que é usada para analisar os pontos fortes e fracos, oportunidades e ameaças de um projeto ou empresa. Aplicada ao mercado de aplicativos, essa análise pode servir para uma análise competitiva ou benchmarking com os aplicativos de concorrentes potenciais.

Usando este método para analisar o aplicativo, fazem-se perguntas para cada um dos quatro componentes que o caracterizam:

- **PONTOS FORTES:** quais são as vantagens do aplicativo que se está a construir, o que ele tem que outros não têm;

- **PONTOS FRACOS:** o que pode ser melhorado ou evitado, quais são os consumidores ou mercados que conseguem identificar os pontos fracos deste aplicativo;
- **OPORTUNIDADES:** quais e quantas oportunidades existem e como pode aproveitá-las, quais as possibilidades de internacionalização da App e quais as tendências de mercado a seguir;
- **AMEAÇAS:** Quais são os possíveis obstáculos, o que os concorrentes podem fazer para criar dificuldades e o que pode acontecer se as tecnologias mudarem. Assim que a análise do aplicativo for concluída, pode usar-se o mesmo método para analisar a oferta do concorrente. No final desta análise, os resultados são analisados e eventuais fragilidades identificadas são corrigidas, reforçando as fortalezas e eliminando eventuais ameaças.

Tão importante como fazer a análise de concorrentes é a escolha da plataforma sobre a qual desenvolver o projeto, escolhendo se se deseja criar um aplicativo para iOS, dispositivos Android (que cobrem uma participação de mercado de 99% em relação a outros concorrentes), ou propor soluções híbridas multiplataforma. É necessário, portanto, analisar os dois principais sistemas operacionais e entender qual é o melhor para o desenvolvimento de aplicativos na área educacional. Uma análise interessante foi desenvolvida por Alexander Findlay, Consultor de TIC, que analisa a introdução da tecnologia móvel nas escolas e compara as duas soluções principais: Apple e Android. A partir dessa análise, a escolha recaiu sobre a tecnologia Apple por diversos motivos: apesar do custo mais alto em comparação com a concorrência do Android, um produto Apple é mais fácil de gerir, independente do tipo de dispositivo (iPad Mini, iPad Air ...) todos irão funcionar com o mesmo tipo de sistema operacional (ao contrário do Android que possui vários sistemas operacionais) e raramente precisará da ajuda de um técnico para resolver problemas.

O uso de aplicativos ou ferramentas digitais para a aprendizagem ainda não é particularmente desenvolvido nas escolas. Existem poucas aplicações como auxílio didático e muito poucas com relação à educação ambiental. Na Fig.6, analisamos as vantagens e limitações de um aplicativo deste tipo.



Fig. 6 Exemplo Analise SWOT de um aplicativo educacional para ensino de educação ambiental criado durante o projeto desenvolvido durante o mestrado em design de interação

3.4 Arquitetura do sistema

Projetar um produto para ser útil deve ser um processo estruturado e sistemático, começando com objetivos de alto nível e avançando para objetivos específicos. Não é possível alcançar uma meta - usabilidade ou outra - se ela permanecer obscura e mal concebida. O termo usabilidade em si também deve ser definido com a organização.

Uma definição operacional do que torna o produto utilizável pode incluir:

- Utilidades
- Eficiência
- Eficácia
- Satisfação
- Acessibilidade

(J. Rubin, D. Chisnell, 2008).

Portanto, a usabilidade digital não termina com a “facilidade de uso”, que continua sendo o

objetivo fundamental das boas práticas de web design.

A usabilidade digital é “fluidez, sensação de controle, sensação de prazer dado pelo rápido alcance dos objetivos que estão sendo perseguidos, descobrindo novas possibilidades, um sentimento de admiração, surpresa e, por que não, uma sensação de espanto”. (M. Visciola, 2006)

Antes de prosseguir com o design de qualquer tipo de tecnologia, será fundamental entender quem é o utilizador.

A maioria das pessoas consideram as pessoas que interagem diretamente com o produto como utilizadores, mas pode ser preciso de considerar outras pessoas.

Isso não significa que tenha que conduzir atividades de análise de utilizadores com utilizadores secundários e terciários, mas pelo menos deve-se saber quem eles são. Se os decisores terciários não comprarem o produto, os utilizadores primários nunca o usarão. (K. Baxter, C. Courage, K. Caine, 2005)

Testes de usabilidade são sessões de observação direta da interação entre um utilizador e um serviço digital.

Os testes são realizados atribuindo ao utilizador uma ou mais tarefas a serem executadas e analisando seu comportamento ao completá-las. Para produzir bons resultados, as sessões de teste devem ser planeadas e organizadas, definindo numa fase preliminar o número e tipo de utilizadores envolvidos, a forma como o teste é realizado, etc. Durante cada sessão, um pesquisador (ou facilitador) apoia o participante, descrevendo as tarefas a serem realizadas, observando a interação de forma direta (sem interrupção) e aguardando o final de cada operação para aprofundar com outras questões. A gravação dos resultados de cada sessão é necessária para preparar um relatório final da atividade, que é um documento que destaca os aspectos funcionais e / ou críticos da experiência de uso atual com base nos dados coletados. Os testes de usabilidade podem ser realizados durante a fase de desenvolvimento de uma plataforma digital, para verificar o comportamento do utilizador e fazer alterações durante a construção. Ou testes de usabilidade podem ser realizados quando um serviço já está sendo testado ou usado, para

entender quais problemas o utilizador encontra em uso e redesenhar alguns componentes da plataforma.

A tarefa dos testes é estudar o comportamento de utilizadores reais lidando com produtos reais ou com seus protótipos, com dois objetivos:

- Identificar problemas críticos para corrigi-los durante a fase de design
- Entender como se move o utilizador e razões para levá-lo em conta na fase de projeto.

Os utilizadores estão sempre cheios de surpresas. Eles percebem conexões que nós nunca previmos. Às vezes isso pode inverter a nossa organização inicial e levar a uma melhor. O mais útil de todos, no entanto, é a espiada dentro dos modelos mentais de utilizadores do conteúdo do site. (Dave Rogers)

4. Processo de Design

Durante a fase de construção do protótipo apresentada anteriormente, foi dada particular importância ao aspecto visual do aplicativo e às escolhas que levaram ao uso de certas fontes e cores que podem ser úteis para a criação de aplicativos para o ensino de crianças e muito mais.

Criar atividades com objetivos claros e explícitos, de forma a ser imediato e intuitivo em cada cena é relevante. Quanto mais importante for uma ação, mais o elemento interativo deve ser evidente e deve ser claramente distinguido do resto da cena. Este efeito pode ser criado usando diferentes profundidades de cor e pequenas animações.

As crianças são mais imaginativas e físicas do que os adultos e é importante tomar nota disso. Para desenhar um aplicativo educacional para crianças é necessário aliar aprendizagem à diversão. Consistência é a chave para ganhar a confiança do utilizador final e para facilitar a navegação numa plataforma de Aprendizagem Digital. No nível da interface gráfica, isso se traduz em cores brilhantes, fontes de fácil leitura e facilidade de uso.

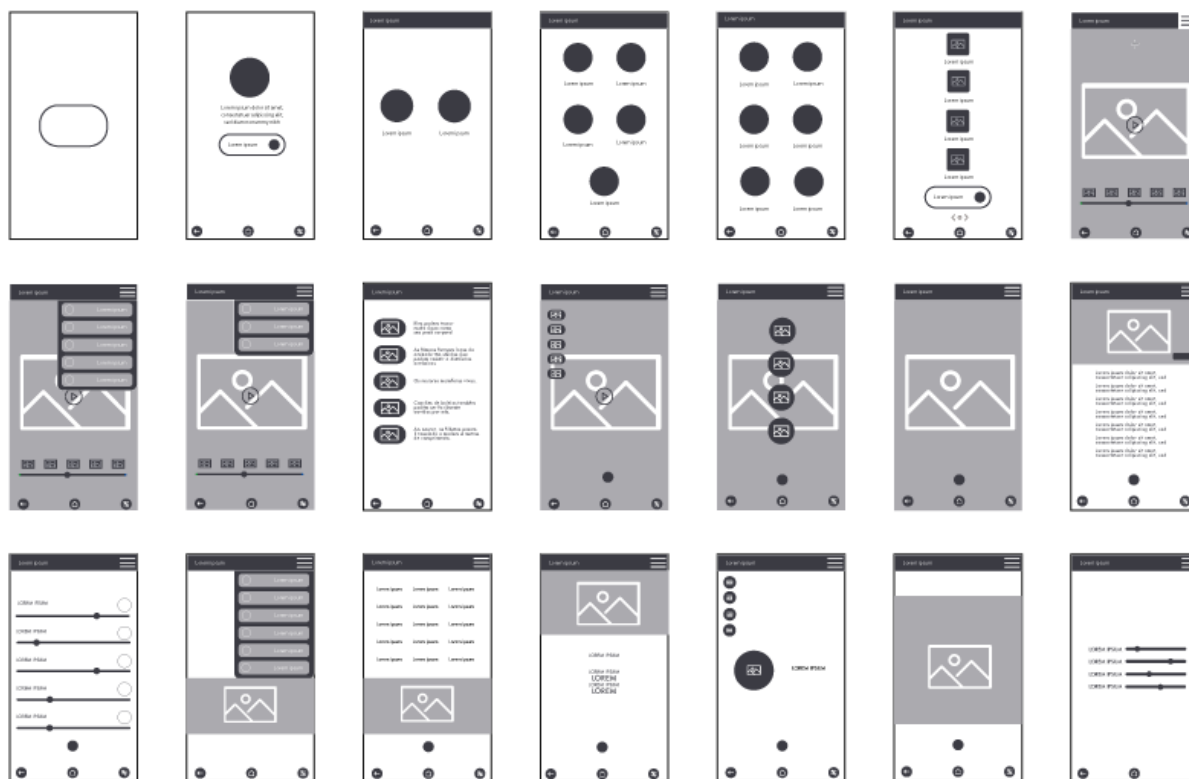


Fig. 7 Exemplo Wireframe criado durante o projeto desenvolvido durante o mestrado em design de interação

4.1 Wireframe

Os wireframes são ilustrações organizacionais esquemáticas do conteúdo do projeto que estamos criando. A principal função dos wireframes é comunicar a ideia do projeto, concentrando a atenção na arquitetura, e não no design. Ele representa uma base para o protótipo ser desenvolvido e dá uma boa ideia da consistência do projeto. Os wireframes são partes do projeto. No entanto, continua sendo um documento em constante evolução em todo o projeto. Principalmente as interfaces principais de primeiro e segundo nível são descritas, mas diferentes aspectos devem ser considerados, incluindo a importância da página, a complexidade da navegação e o tempo.

Graças aos wireframes, podemos identificar a dinâmica do projeto em diferentes áreas: da usabilidade à navegabilidade e ao design (em termos de estrutura e organização do conteúdo).

Para os wireframes do sistema iHab, optamos por uma estrutura simples e bem organizada. Botões grandes e facilidade de uso são os principais recursos do sistema. Além disso, como é evidente, imagens grandes prevalecem para atingir o utilizador e deixá-lo pode excitar.

4.2 Linguagem visual



Fig. 8 Exemplos linguagem visual criadas durante o projeto desenvolvido durante o mestrado em design de interação

Conhecendo agora a estrutura técnica do sistema, teremos que lidar com a criação da interface gráfica do utilizador (GUI). A interface gráfica é o coração pulsante do design do aplicativo, graças ao qual a aparência geral do aplicativo é apresentada. Neste caso, o design de uma aplicação inclui muito mais do que apenas fontes e cores. Na verdade, a interface gráfica completa, incluindo funções e elementos de controle, faz parte do design do aplicativo. Por isso, é importante ter um uso bem pensado de todos os componentes gráficos do aplicativo e determinar sua usabilidade são, acima de tudo, a posição, o tamanho e a estruturação do conteúdo do aplicativo.

Assim, o termo “layout” indica a estrutura visual da interface do aplicativo e o layout do conteúdo do aplicativo.

Embora imagens e símbolos predominem no layout, alguns segmentos do aplicativo exigem uma realização textual. Especialmente para os elementos que determinam decisivamente a aparência de uma interface gráfica (menus, artigos, etc.), o uso de um carácter apropriado é muito importante.

As cores

As cores a serem utilizadas dependem do tema e do destino da aplicação, mas é preciso lembrar que um contraste bem pensado facilita a leitura do conteúdo. Após os 11 anos, a criança descobre as variações nas cores dos objetos.

Caso o aplicativo não envolva muito texto, seja uma experiência mais envolvente, recomenda-se um fundo escuro para permitir que o utilizador relaxe e torne a experiência mais confortável.

Texto

Ao desenvolver aplicativos voltados para crianças, evite evitar-se rótulos textuais e textos longos. É melhor concentrar-se nas instruções áudio combinadas com o destaque do texto durante a narração. As fontes utilizadas devem ser sans-serif e facilmente legíveis, os parágrafos devem ser curtos e o presente deve ser usado como forma verbal.

Normalmente, para aplicativos iOS, a fonte "San Francisco" era usada pelo iOS 9, enquanto anteriormente "Helvetica" e "Helvetica Neue" eram as fontes do sistema para aplicativos iOS. Em vez disso, a fonte padrão para o sistema Android é a versão 4.0 "Roboto".

Para um aplicativo educacional para crianças, também se deve considerar uma fonte clara e



Fig. 9 Exemplos cores

legível e a “*Futura*” é excelente. *Futura* é uma fonte projetada por Paul Renner em 1927. It é

uma fonte sans serif (sem agradecimento). No desenho das letras é possível reconhecer uma geometria rigorosa e um desenho racionalista que se tornou representativo da escola Bauhaus. Na verdade, foram utilizadas figuras geométricas puras para desenhar o *Futura*: o quadrado, o triângulo e o círculo. *Futura* é uma fonte opaca, geométrica, harmoniosa e linear, portanto um carácter distintivo, onde a clareza e a omissão de todos os elementos supérfluos são os princípios básicos da sua estrutura geométrica.



Fig. 10 Type Guide por Futura de Type-Trap []

Dimensão

O polegar é a unidade de medida de um aplicativo, pois é principalmente com o polegar que controlamos os dispositivos. O desenho dos elementos gráficos deve, portanto, levar em consideração o fator "dedo gordo" e evitar botões muito pequenos ou muito próximos. Considerando o fator "dedo gordo", sobre o qual falamos acima, o tamanho mínimo de um botão é de 44 px. No entanto, o tamanho não é a única variável; O layout dos botões também é muito importante, pois a proximidade excessiva aumenta as chances de selecionar o comando errado, reduzindo a usabilidade do app.

Iconografia

O ícone é um símbolo gráfico cuja visualização recupera, da memória de curto ou longo prazo, lembranças relacionadas a vários fatores. Um ícone pode representar objetos, processos e ações/ operações desde que corretamente projetado. Ele irá ajudar (e muito) na interação humano-computador pois o utilizador irá identificar e escolher de forma rápida, fácil e direta a ação que deseja executar no sistema.

Para sistemas destinados a crianças, acreditamos que seja apropriado projetar dois tipos de ícones:

- ícones principais
- ícones secundários

Os ícones principais são os ícones reais da interface.

Estes permitem certas operações, tais como:

- Voltar para a Home
 - Ir para as configurações
 - Abrir o menu
- Voltar
- Pesquisa
- Ouvir

Ícones secundários ajudam a entender o texto ou o conteúdo para ajudar o utilizador a atingir seus objetivos.

Áudio

Os efeitos sonoros dão vida à cena, estabelecem o clima da aplicação e fornecem um feedback útil ao utilizador. Os sons devem ser consistentes com a ação à qual estão conectados e o volume

geral deve ser moderado. Com a presença de um microfone integrado em tablets e dispositivos móveis, também é possível gravar a entrada do utilizador que pode ser usada para ativar comandos ou simplesmente replicar frases ouvidas anteriormente. Quando um som é gravado através do microfone, é aconselhável sinalizar o reconhecimento da entrada com um efeito sonoro.

Interatividade

Usar personagens familiares às crianças. Os aplicativos devem incluir um ou mais personagens principais que transmitam uma sensação de amizade e diversão. Esses personagens são como anfitriões que têm a tarefa de guiar a criança ao longo do processo de aprendizagem. Esse é um aspecto fundamental porque todas as experiências que se seguem serão construídas a partir dessa relação. A maior parte do conteúdo deve ser apresentada por um personagem que atua como narrador e explica o propósito do ambiente interativo. O próprio narrador tem a função de facilitar e envolver o utilizador sempre que ele não interagir com a app. Em aplicações onde a interatividade está presente, é útil inserir um cronômetro que, após alguns segundos de inatividade, avisa o utilizador para responder à atividade. O personagem principal também fornecerá feedback para resultados positivos e negativos. Caso o utilizador não consiga atingir a meta após uma primeira sugestão, é aconselhável inserir um segundo nível de informação, também fornecendo feedback para estimular a atividade. Também é importante variar o tipo de atividade dentro da app, passando assim dos processos de leitura aos jogos, sempre fornecendo feedback de áudio ou vídeo que reforça as ações corretas.

Quanto mais simples de aprender for uma interface, mais rápido o utilizador se dedicará ao conteúdo, mantendo baixos os limites de atenção em todas as ferramentas de acessórios. Também é útil inserir uma Ajuda contextual que, somente se solicitada, fornece ao utilizador informações sobre as ações permitidas, sugere os tipos de gestos permitidos e orienta o utilizador para a solução da tarefa.

Gestos

Todas as ações e gestos que o utilizador realiza na tela do dispositivo. Além de ter um conteúdo bem desenhado e consistente com o alvo, os gestos que são feitos na tela podem ajudar a fixar um conceito ou entender algo. Os gestos devem ser escolhidos para fortalecer o vínculo entre conteúdo e interação, mas também para permitir a manipulação direta dos elementos na tela. Entre os gestos mais intuitivos podem ser listados:

Tap: único toque da tela, que lembra um clique do mouse. O gesto mais difundido e compartilhado em todos os dispositivos de toque.

Drag: arraste seu dedo na tela. As crianças adoram desenhar e traçar na tela; no entanto, eles têm dificuldade em traçar caminhos muito longos ou formas geométricas precisas. Portanto, é útil usar sistemas de preenchimento automático que detectam o caminho traçado pela criança e avaliam se a tarefa foi executada ou não.

Swipe: deslize rapidamente o dedo na tela. Deslizar o dedo pela tela pode ser usado para virar as páginas virtuais. No entanto, continua sendo importante indicar a direção em que essa ação é permitida.

Entre os gestos menos intuitivos podemos encontrar Pinch, Double Tap ou Multitouch, por apresentarem problemas críticos justamente pelo uso simultâneo de vários dedos na tela.

4.3 Criação de interface gráfica

Chega-se então à fase de colocar em prática o conhecimento adquirido fazendo o protótipo de um aplicativo. Seguindo as regras de design listadas acima, podemos passar para a fase de prototipagem, para dar vida às ideias que nasceram durante as fases de pesquisa e análise explicadas nos capítulos anteriores. Já foram identificados os alvos e as disciplinas da prototipagem, ou seja, a criação de aplicativos para o ensino de educação ambiental para alunos do ensino médio entre 11 e 12 anos e análise de concorrentes já existentes no mercado.

Numa primeira fase são criados protótipos com aplicações como Adobe XD, Invision ou Sketch para enfrentar os primeiros testes de usabilidade.

Jakob Nielsen define a usabilidade como uma medida da qualidade da experiência de um utilizador em interação com algo, seja um website ou um aplicativo de software tradicional ou

qualquer outra ferramenta com a qual o utilizador possa operar. Segundo o Nielsen, um produto é utilizável quando é fácil de aprender, permite uma eficiência de uso, é fácil de lembrar, permite poucos erros de interação e de baixa gravidade, é agradável de usar.

Em resumo, portanto, para ser utilizável, um produto deve:

- ser adaptado às necessidades e expectativas dos utilizadores finais específicos que o utilizam sob certas condições;
- seja fácil de entender, aprender, usar e ser agradável;
- permitir realizar as atividades específicas do trabalho de forma correta, rápida e com satisfação.

Esses três pontos, além de resumirem os diversos aspectos que as diferentes definições de usabilidade destacam, colocam o acento em algumas etapas e atividades a serem realizadas durante o ciclo de produção de um site, a fim de garantir a usabilidade do produto.

De fato, para que um produto seja adaptado às necessidades e expectativas dos utilizadores finais, é necessário conhecer bem as características dos utilizadores, as atividades que realizam e o contexto organizacional e social em que estão inseridos e operam.

No que diz respeito à facilidade de entendimento e uso, atualmente, princípios e diretrizes de interface consolidados estão disponíveis, baseados no conhecimento dos aspectos cognitivos e comportamentais da interação humano-computador, o que pode facilitar a criação de um produto utilizável.

No entanto, esses aspectos não são suficientes. Para garantir a usabilidade, é fundamental comparar-se constantemente os utilizadores finais, com aqueles que estarão diretamente operando no produto uma vez disponível, para verificar, avaliar e, possivelmente, medir as escolhas de design em termos de satisfação real do utilizador.

Testes de usabilidade são sessões de observação direta da interação entre um utilizador e um serviço digital. Os testes são realizados individualmente, atribuindo ao utilizador uma ou mais tarefas a serem realizadas, e analisar seu comportamento ao completá-las.

Para produzir bons resultados, as sessões de teste devem ser planeadas e organizadas em detalhes, definindo em uma fase preliminar o número e o tipo de utilizadores envolvidos, o procedimento de teste, o método de medição escolhido e as tarefas a serem verificadas.

A gravação dos resultados de cada sessão é necessária para elaborar um relatório final da atividade, ou um documento que destaque os aspectos funcionais e / ou críticos da experiência de uso atual com base nos dados coletados. Testes de usabilidade podem ser realizados durante a fase de desenvolvimento de uma plataforma digital, para verificar o comportamento dos utilizadores e fazer alterações durante o trabalho; ou testes de usabilidade podem ser realizados quando um serviço digital já está sendo testado ou em uso, para entender quais problemas o utilizador encontra ao usar e redesenhar alguns componentes da plataforma de acordo.

O teste de utilizador on-line ocorre remotamente: pode ser moderado, ou seja, o observador está indiretamente presente - por meio de ferramentas de videoconferência, por exemplo - ou pode ser não moderado e, neste caso, não há observador humano, mas um software que registra os movimentos mouse e respostas do utilizador e organizar os dados coletados para facilitar a leitura.

O teste de utilizador on-line é certamente menos informativo do que se for presencial, mas certamente mais barato e mais rápido. As áreas de aplicação dos testes em presença e on-line são - na minha opinião - diferentes, portanto, não são propostas como ferramentas alternativas, mas como complementares.(K. Soucy 2010)

“Knowing how people will use anything is fundamental to the design and the future use of that same thing.”
[Donald Norman]

4.4 Unmoderate test

O teste de usabilidade não moderado permite testar sessões com centenas de pessoas ao mesmo tempo, em seu ambiente natural, que, por sua vez, fornece dados quantitativos e também alguns

dados qualitativos. As métricas e o feedback exatos que pode-se recolher variam de acordo com a ferramenta usada.

A criação e gestão de um estudo de usabilidade unmoderated é semelhante ao processo de criação e administração de um inquérito online, mas com os passos adicionais de estúdio tradicional da usabilidade, como se segue:

- Defina o estudo. Decida as quais tarefas necessárias para que os participantes executem, a ordem das atividades e que perguntas de acompanhamento se deseja perguntar sobre a sua experiência.
- Recrute participantes. O participante do teste é tão importante para um teste de usabilidade não moderado quanto para um teste moderado. A equipa baseará importantes decisões de design nos dados obtidos, de os participantes que são utilizadores reais mas também potenciais de um produto.
- Inicie o teste e envie convites por email.
- Analise seus resultados. A maioria das ferramentas de teste não moderadas oferece relatórios em tempo real durante os testes.

Existem vários tipos de testes modernos que podemos encontrar online.

A plataforma UsabilityHub oferece um conjunto completo de ferramentas de teste que permitem descobrir rapidamente problemas de design, evitando desperdício de tempo, esforço e frustração do utilizador. ()

Graças a esta plataforma podemos analisar e avaliar tanto os métodos de navegação dos utilizadores quanto as suas preferências ou problemas gráficos.

4.5 Moderate test

Testes de usabilidade são como muitas coisas, fáceis de fazer, mas difíceis de fazer e melhorar apenas através da prática.

A maneira como o moderador interage com os participantes, primeiro recrutando-os e depois agradecer no final da sessão, é essencial para o sucesso do teste, a validade dos dados e a

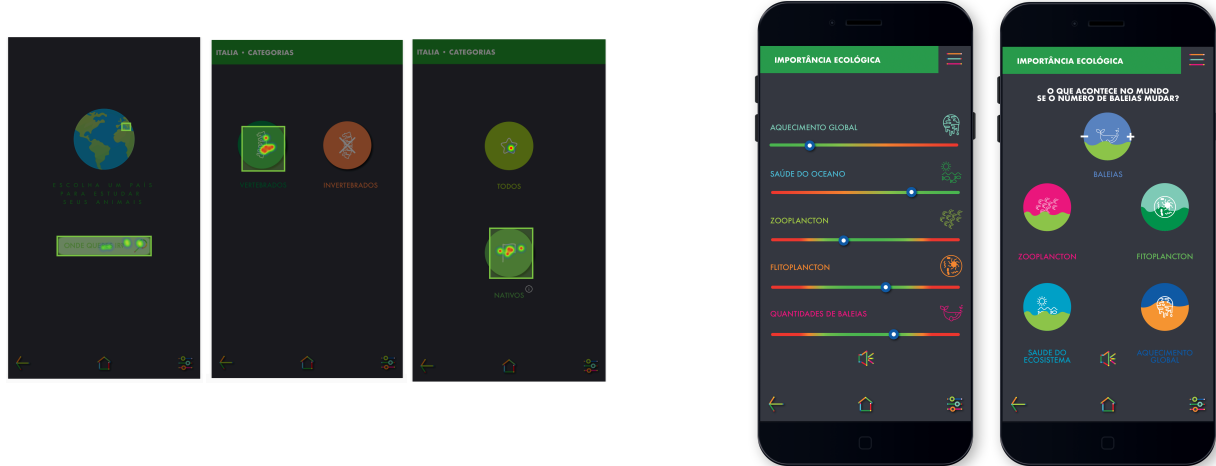


Fig. 11 Exemplos Unmoderate Test criados durante o projeto desenvolvido durante o mestrado em design de interação

reputação da organização do moderador.

Um moderador do teste deve ser imparcial e neutro em relação ao produto, sendo aberto e acessível aos participantes. Essa combinação de neutralidade e acessibilidade pode ser difícil de alcançar.

De acordo com Joseph S. Dumas e Beth A. Loring, em seu manual "Moderating Usability Tests, Principles and Practices for Interacting"(2008), as regras de ouro para um bom teste de usabilidade são:

1. Decidir como interagir com base no objetivo do teste.
2. Respeitar os direitos dos participantes.
3. Lembrar-se de sua responsabilidade para com os futuros utilizadores.
4. Respeitar os participantes como especialistas, mas mantendo-se no seu cargo.
5. Ser profissional, o que inclui ser genuíno.
6. Deixar os participantes falarem!
7. Lembrar-se que sua intuição prejudicá-lo ou ajudá-lo.
8. Seja imparcial.

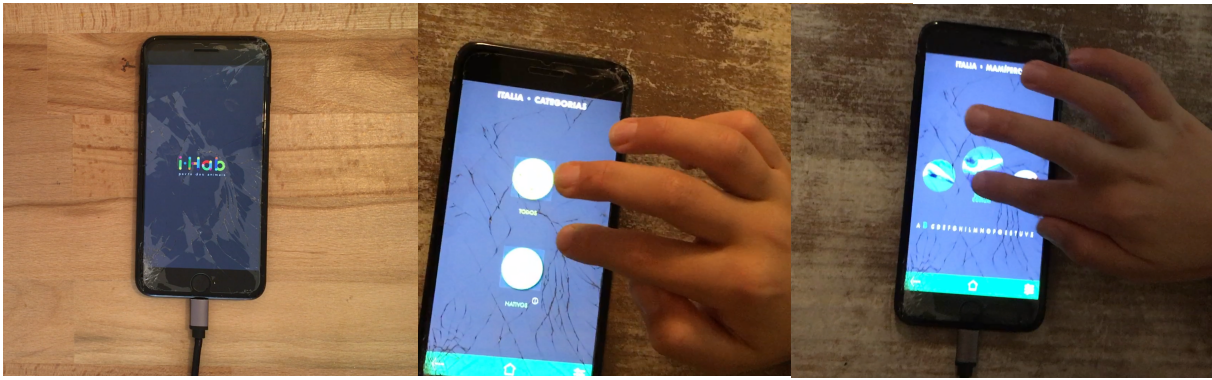


Fig. 12 Exemplo de funcionalidade teste de Think-Aloud criado durante o projeto desenvolvido durante o mestrado em design de interação

9. Não dar informações inadvertidamente.

Um teste moderado muito útil com crianças é o Think-Aloud. Sessões moderadas de teste com crianças e adolescentes podem ser divertidas e recompensadoras. A primeira coisa essencial a fazer é pedir permissão aos pais e informá-los sobre o tipo de teste para que enviaremos aos filhos. Esta fase é fundamental, necessária e obrigatória enquanto para menores.

Graças a esses testes é possível destacar imediatamente à primeira vista a facilidade e naturalidade com que as crianças usam os dispositivos eletrônicos, utilizando gestos como deslizar, clicar ou arrastar sem problemas, reconhecendo quais os botões que realizaram uma ação. Graças à presença de ícones e áudio, as crianças sabem o que fazer e como se mover.

5. Prototipagem

Numa primeira fase o sistema é prototipado com ferramentas como invision, sketch ou Adobe XD, para enfrentar os primeiros testes de usabilidade.

Tradicionalmente, os designers usavam o Photoshop para criar modelos de aplicativos e sites, mas o Photoshop como ferramenta nunca foi criado com o propósito de projetar sites e aplicativos. O grande problema com o Photoshop é que a ferramenta é desajeitada, seu formato

PSD nativo impede que os designers lidem com arquivos grandes e a maioria das ferramentas do software são inúteis para a maioria dos designers digitais.

O Sketch apareceu e tratou dessas questões e agora evoluiu para se tornar o líder de mercado. O Sketch permite símbolos reutilizáveis e fluxos de trabalho muito mais rápidos, permitindo que os designers trabalhem de maneira mais inteligente e eficiente. O verdadeiro poder do Sketch vem à tona com a adição de plugins, que permitem adicionar recursos muito específicos e personalizados ao software, não disponíveis por meio do aplicativo nativo.

A capacidade de criar protótipos foi possível pela primeira vez por meio da integração do InVision. O InVision permite que você carregue arquivos nativos PSD e Sketch para sua plataforma e transforme projetos em protótipos viáveis. O Sketch recentemente possibilitou a prototipagem por meio de seu serviço de nuvem.

Quando se tratava de animações complexas, os designers frequentemente ainda dependiam da prototipagem em softwares como o After Effects da Adobe. Esse fluxo de trabalho foi posto à prova quando a InVision anunciou o Studio, um novo software que permitia aos designers criar animações e microinterações mais avançadas. O Studio também integrou a vinculação do InVision ao Sketch por meio de seu plug-in Craft. Como o Photoshop, o After Effects é um aplicativo muito pesado para o trabalho de animação da IU.

Os designers da Adobe viram a necessidade de uma nova ferramenta de design e responderam com um novo software chamado "Project Comet" em outubro de 2015. A ferramenta assumiu uma nova filosofia de design e prometeu ao designer a capacidade de “projetar na velocidade do pensamento”. O Comet foi lançado com um novo conceito fantástico e uma ferramenta chamada repeat-grid, que chamou a atenção de muitos designers.

Isso permitiu que os designers criassem um elemento e, em seguida, usassem a ferramenta de repetição de grade para arrastar uma grade de instâncias do mesmo elemento de maneira semelhante a uma grade. Texto e imagens podem mudar dentro da grade, mas os elementos permaneceram conectados, permitindo redesenhos em cascata ajustando um elemento,

economizando horas para os designers com a capacidade de experimentar layouts mais rápido do que nunca.

Quando o Comet saiu do beta, foi renomeado para Adobe XD (XD que significa Experience Design). Apesar de ser uma ferramenta muito poderosa, ainda não conseguiu competir com o Sketch e seu ecossistema de plugins. A prototipagem era básica, na melhor das hipóteses, mas a prototipagem via InVision era melhor.

Tudo isso mudou quando a Adobe lançou sua atualização mais recente para o XD em 15 de outubro no Adobe MAX 2018.

O Adobe XD também permitiu aos utilizadores interagir com protótipos como nunca antes. Gatilhos de voz e reprodução de voz permitem que você leve os utilizadores além da tela, permitindo a capacidade de prototipar com voz. Algo que nenhum outro software de prototipagem permite fazer.

A capacidade de ir de wireframe a prototipagem de alta fidelidade e etapas de desenvolvimento numa peça de software mostra o quanto a Adobe se concentra em fluxos de trabalho. Ao tornar os fluxos de trabalho simples e fáceis, está a abordar-se alguns dos maiores problemas dos designers, algo de que somente a Adobe é realmente capaz por meio de seu ecossistema de software.

Numa segunda fase, prototipagem e programação podem se cruzar e uma excelente ferramenta usada por designers para abordar este segundo momento é o Processing.

Processing é uma linguagem de programa visual criada em 2001 por Ben Fry e Casey Reas no MIT Media Lab em Boston. O objetivo do Processing é criar uma linguagem que possa ser usada por artistas visuais para experimentar novas formas de comunicação e, dada a sua simplicidade, como um hardware de aprendizagem para aproximar as pessoas da programação. Visual é, na verdade, o seu termo chave porque, ao contrário da maioria das outras linguagens, Processing também permite que aqueles que nunca escreveram uma linha de código tenham feedback visual imediato do seu trabalho na tela do computador.

5.1 Ambiente software Processing

Já em 1945, mas também muito antes, as primeiras linguagens de programação começaram a surgir.

A primeira linguagem de programação da história é a linguagem mecânica usada por Ada Lovelace para programar a máquina de Charles Babbage, seguida pelo Plankalkül de Konrad Zuse, desenvolvido por ele na Suíça neutra durante a Segunda Guerra Mundial e publicado em 1946. O Plankalkül nunca foi realmente usado para programação .

Milhares e milhares de idiomas diferentes foram criados desde então.

A maioria das linguagens de programação posteriores tentaram afastar-se do nível básico, dando a possibilidade de representar estruturas de dados e estruturas de controle que são mais gerais e mais próximas da forma (humana) de representar os termos dos problemas para os quais pretendemos para escrever programas.

Existem basicamente três maneiras de catalogar uma linguagem de programação:

Geracional:

- primeira geração: as várias linguagens de máquina proprietárias, decididamente complexas e fortemente vinculadas à arquitetura de hardware específica, caem nesta categoria;
- segunda geração: constituem uma primeira abstração da linguagem de máquina (assembly), permitindo ainda melhorar a atividade de desenvolvimento graças ao uso de um número menor de instruções. Com eles vêm os montadores, ou seja, programas capazes de traduzir a linguagem assembly em linguagem de máquina;
- terceira geração: as linguagens que mudam o foco para a linguagem natural (como o inglês) enquadram-se nesta categoria, tornando-as muito mais intuitivas do que a linguagem de montagem e de máquina. Essas linguagens também são definidas como linguagens de alto nível e entre elas estão incluídas em C, C ++, Basic, Java, C #, etc...;
- quarta geração: nesta categoria encontramos as linguagens declarativas, cujo representante mais ilustre é o SQL, utilizado principalmente em bancos de dados relacionais.

Execução:

- compilados: são as linguagens cujo código, para ser executado, é traduzido para a linguagem de máquina específica do computador em que será executado. Esta operação é chamada de compilação e é realizada por um compilador (por exemplo: gcc e g ++). Alguns exemplos de linguagens compiladas são C e C ++;
- interpretados: o código não é compilado, mas é interpretado em tempo de execução em conjunto com os dados de contexto. Entre elas estão as linguagens de Scripting como: PERL, PHP, etc...;
- compilação intermediária: neste caso existe uma situação híbrida entre as duas anteriores. O código é primeiro compilado em uma linguagem intermediária e, posteriormente, interpretado por um mecanismo especial que leva o nome de Máquina Virtual ou Runtime. Os exemplos mais relevantes são .Net (.Net Runtime) e Java (Java Virtual Machine, JVM). Na realidade, o cenário definitivamente evoluiu e mais do que uma interpretação, na segunda fase ocorre uma compilação Just-inTime (JIT) para o sistema específico;

Paradigma de programação:

- procedural: o código é dividido em sub-rotinas (funções e / ou procedimentos) que realizam tarefas específicas, geralmente permitindo tirar vantagem de saltos condicionais perigosos (também conhecidos como go-to). A divisão em funções (ou procedimentos) visa reduzir a complexidade do programa como um todo, adotando a estratégia de dividir para conquistar e abarcar uma abstração funcional em que basta conhecer apenas a interface do procedimento, restando os detalhes de implementação. A definição "procedural" deriva da linguagem COBOL, que foi a primeira a usar este conceito;
- estruturado: na prática, uma programação procedimental purificada de saltos condicionais. A possibilidade de realizar tudo o que é possível com linguagens procedimentais sem o uso de saltos condicionais perigosos é demonstrada no teorema dos dois matemáticos italianos Böhm-Jacopini: *qualquer algoritmo pode ser transformado*

em um algoritmo equivalente composto apenas de combinações de sequências, seleções e iterações (eliminando assim a necessidade de saltos condicionais). Algumas das linguagens de programação mais famosas enquadram-se nesta categoria: de Pascal a ADA a C;

- orientado a objetos (OOP - Object Oriented Programming): neste caso, o fulcro de todo o programa é a classe (e sua interface) composta de atributos funcionais e métodos. A instância de uma classe é definida como um objeto ou uma entidade com um estado específico. Uma linguagem a ser definida como "orientada a objetos" deve contemplar três propriedades fundamentais: encapsulamento, herança e polimorfismo. O objetivo é atingir a chamada abstração de dados, com foco em informações essenciais. Entre as linguagens mais famosas encontramos: C ++, Java, C #, etc...;
- declarativa: neste paradigma, a programação é desenvolvida usando elementos característicos da esfera matemática:
 - a programação funcional, é baseada no uso de funções matemáticas. A linguagem mais conhecida é o LISP, muito usado no campo da inteligência artificial;
 - A programação lógica, é baseada na lógica de primeira ordem para representar e processar informações, e é dividida em programação lógica indutiva e programação lógica abdutiva. A linguagem mais estabelecida nesta área é o Prolog.

Em 2001, o Instituto de Tecnologia de Massachusetts (MIT) patrocinou a criação do Processing (processing.org) para fornecer uma ferramenta para aqueles que não são realmente cientistas da computação criarem pequenos aplicativos.

A primeira versão do Processing foi criada em 2001, a versão mais avançada foi criada em 2004, mas é atualizada periodicamente.

O núcleo do Processing foi, ao longo dos anos, enormemente enriquecido com bibliotecas que permitem gerir câmeras, microfones, áudio, interações, mouses, etc.

Processing é filho de uma experiência realizada alguns anos antes, também no MIT de Boston, pelo grande designer e arquiteto John Maeda.

John Maeda havia proposto uma ferramenta chamada DBN (Design by Numbers) com a qual os utilizadores, escrevendo pequenas linhas de código, eram capazes de criar ladrilhos coloridos de formas geométricas e abstratas. A ideia cultural desta ferramenta foi implementar no computador um daqueles mitos que estão na base da arte minimalista: o que é uma obra de arte? O produto do artista ou as ações do artista? O resultado ou como faço para chegar lá? No DBN, a obra de arte era tanto o cartão colorido que aparecia na tela, como a sequência de instruções que o utilizador ia escrever.

No entanto, era uma ferramenta muito pobre e muito trabalho foi necessário para desenvolver trabalhos mais interativos. Por este motivo foi criado o Processing.

Processing é uma extensão da linguagem Java da qual herda muitas formas de escrita, muitas construções e sintaxe básica.

Quando se executa o Processing, na verdade está a executar um programa Java. O processamento permite que se escreva um código muito mais simples do que se precisaria escrever se se estivesse trabalhando em Java puro. Também permite que se estruture o código num estilo baseado em função (procedural) e não orientado a objetos. Mas, depois de compilar o código, a saída é convertida em arquivos de classe Java - indistinguíveis de qualquer outro programa Java - e os arquivos de classe são interpretados dentro da Java Virtual Machine à medida que o programa é executado. (I. Greenberg 2006)

5.2 Biblioteca P5.js

Um dos principais objetivos do Processing (e dos projetos baseados em Processing) é tornar o mais fácil possível para os iniciantes aprenderem a programar aplicativos gráficos interativos (enquanto fornece ferramentas poderosas para especialistas).

P5.js é uma biblioteca JavaScript, uma plataforma online que é um desenvolvimento do Processing e que também integra a linguagem JavaScript. p5.js é uma interpretação de Processing para a web atual.

O editor da web p5.js foi desenvolvido desde 2016 sob a orientação de Cassie Tarakajian.

A vantagem de usar a linguagem de programação JavaScript é sua ampla disponibilidade e suporte onipresente: cada navegador da web tem um interpretador JavaScript embutido, o que significa que os programas p5.js podem (geralmente) rodar em qualquer navegador da web. Além disso, a linguagem JavaScript é definida por um padrão internacional e a maioria dos intérpretes JavaScript (incluindo aqueles executados em navegadores da web) são de código aberto e estão disponíveis gratuitamente. Ninguém “possui” JavaScript e qualquer um pode usá-lo gratuitamente.

Nascido em 1995 pela Netscape, empresa que produziu o Netscape Navigator o navegador mais popular do mundo na primeira metade dos anos 90, para dar vida às páginas da Web. JavaScript é mais propriamente uma linguagem de script, ou seja, graças à qual se pode escrever scripts, arquivos que contêm sequências de instruções que devem ser executadas no navegador do utilizador. A execução do script escrito em JavaScript permite visualizar aqueles efeitos legais, divertidos e funcionais que transmitem ao navegador a sensação de movimento e dinamismo que faltava nas páginas escritas na linguagem HTML padrão.

Outra ideia importante do P5.js é que deve ser fácil para os programadores criar protótipos de software muito rapidamente, tentar uma nova ideia ou ver se algo funciona. Por este motivo, os programas de processamento (e p5.js) são geralmente chamados de “sketches”.

Existem várias maneiras de escrever esboços p5.js. Uma vez que p5.js é apenas uma biblioteca JavaScript, pode ser incluído em qualquer página da web normal e até mesmo usá-lo em conjunto com outras bibliotecas JavaScript.(Software libero & TD, 2017)

5.3 Processo de prototipagem com P5

Nos capítulos anteriores, falamos sobre como as ferramentas educacionais digitais podem ser excelentes aliadas para ensinar educação ambiental a crianças.

Acreditamos que essas ferramentas não devem ser usadas da mesma forma que um livro tradicional, mas devem explorar seu potencial para focar no aluno, criando uma experiência quase totalmente imersiva.

Vídeos, interações e estímulos auditivos são a base para a concepção de um aplicativo que visa consciencializar e atrair as crianças sobre a importância de proteger os ecossistemas.

A importância ecológica dos animais, por exemplo, é um assunto pouco falado e do qual há uma imensa necessidade de conhecer.

Por importância ecológica queremos dizer quanto e como um ser vivo está ligado ao ecossistema que o rodeia: como o ecossistema varia de acordo com as variações quantitativas e comportamentais de um determinado ser vivo.

De seguida vai falar-se de um exemplo real que pode ser utilizado para criação de um protótipo, utilizando baleias.

Por falar em importância ecológica, não podemos deixar de citar as baleias como um exemplo concreto. De acordo com um estudo realizado pelo Fundo Monetário Internacional, as baleias podem ser a solução mais verde para as alterações climáticas. Por incrível que pareça, é o resultado de um ecossistema totalmente natural no qual vivem esses cetáceos e que contribuem para o seu próprio funcionamento. Segundo o estudo, as baleias são capazes de absorver uma quantidade de dióxido de carbono igual a cerca de 4 florestas amazónicas.

Com a sua respiração, os cetáceos marinhos são capazes de "respirar" 40% do CO₂ produzido em todo o nosso planeta, ou a impressionante quantidade de 37 mil milhões de toneladas. Falando novamente em números, as baleias absorvem, portanto, o trabalho que deveria realizar 1700 mil milhões de árvores plantadas que, como antecipamos antes, correspondem a aproximadamente a 4 florestas do tamanho da Amazônia. Essa poderosa absorção de que as baleias são capazes é possível graças à sua capacidade de trazer à superfície os minerais do mar que ainda estão no fundo do mar, essenciais para o desenvolvimento do fitoplâncton, o mais

eficiente na sucção de dióxido de carbono. Resumindo, quanto mais baleias existirem, mais CO₂ é absorvido.

Já em 2010, um estudo da Divisão Antártica Australiana destacou a importância do ecossistema de baleias para a absorção de CO₂; de acordo com esta pesquisa específica, suas fezes eram de fato capazes de representar cerca de 12% do ferro na superfície do oceano Atlântico, possibilidade que ao longo dos anos tem sido limitada pela caça indiscriminada desses cetáceos, comprometendo assim a funcionalidade essencial dos ecossistemas que as baleias ajudam a preservar.

Stephen Nicol, autor do estudo de 2010, afirmou peremptoriamente: "Ter até 1% a mais de plancton nos mares significa absorver centenas de milhões de toneladas de CO₂ por ano. Como se, a cada ano, aparecessem repentinamente bilhões de árvores adultas".

Acreditamos que contar, ensinar e fazer as crianças compreenderem essas conexões da natureza é essencial para criar nelas maior consciência e sensibilidade.

Acreditamos também que para a educação ambiental a palavra-chave é "conhecer".

Mas como realmente mostrar essas conexões da natureza de forma simples, intuitiva e imediata?

Os elementos gráficos e sua interação podem nos ajudar.



Fig. 12. Exemplos elementos gráficos que ajudam a compreensão criados durante o projeto desenvolvido durante o mestrado em design de interação

A ideia é que clicar nos ícones "+" e "-" muda a quantidade do número do animal no ecossistema mostrando as mudanças que isso acarreta entre o que ele se alimenta, o que se alimenta dele, a saúde do ecossistema e as alterações climáticas

Já na fase de prototipagem em linguagens de programação, a questão é: como criar uma animação a partir de uma variável que pode ser modificada com um simples clique?

p5.js foi criado para traduzir a linguagem java script para criar imagens, animações e interações. A ideia é escrever uma única linha de código e fazer um círculo aparecer na tela. Processo mais algumas linhas de código e o círculo segue o mouse. Outra linha de código e o círculo muda de cor quando o mouse é pressionado.

Essa ação é chamada “sketching” (esboçar) com o código. Escreva uma linha, depois adicione outra e depois outra, outra e assim por diante. O resultado é que o programa é criado uma parte de cada vez. Esboçar é uma maneira de pensar. (Casey Reas e Ben Fry, 2010)

Nesta linguagem há duas funções muito importantes: `function setup()` e `function draw()`. As operações que serão realizadas apenas na fase inicial são coletadas no "setup()" e aquelas que, ao invés disso, serão constantemente repetidas pelo computador em "draw()".

A ideia básica é criar uma animação baseada em uma variável que pode ser modificada com um simples clique.

O que podemos fazer com p5.js é instruir o programa a criar determinado conteúdo por meio de um certo:

“Se o número do animal varia positivamente, o retângulo interno cresce e fica cada vez mais verde. Se o número do animal varia negativamente, o retângulo interno diminui e fica cada vez mais vermelho ”

Além disso,

“Se o número do animal varia positivamente, o fator X varia positivamente e o retângulo dentro dele cresce e fica cada vez mais verde. Se o número do animal varia negativamente, o retângulo dentro dele diminui e fica cada vez mais vermelho ”.

Esse tipo de abordagem de processamento é chamado de "Conditional Statement" e é definido como: “um conjunto de regras executadas se uma determinada condição for atendida. Às vezes, é referido como uma instrução If-Then, porque, se uma condição for atendida, uma ação será executada.” (Computer Hope, 2017)

Os computadores, infelizmente (ou felizmente), não são máquinas inteligentes e seguem uma lógica bastante simples: uma condição pode ser verdadeira ou falsa, em inglês true ou false.

Uma coisa muito importante é que não pode haver nenhuma variável que seja verdadeira e falsa ao mesmo tempo.

Não podemos esquecer que a aplicação para fazer o protótipo é destinado a um público muito jovem e, portanto, deve ser o mais intuitivo possível

Além disso, para criar o efeito de crescimento e mudança de cor, será necessário criar uma função.

As funções são um dos blocos de construção fundamentais do JavaScript. Uma função em JavaScript é semelhante a um procedimento: um conjunto de instruções que executa uma tarefa ou calcula um valor, mas para um procedimento ser qualificado como uma função, ele deve exigir uma entrada e retornar uma saída onde há uma relação óbvia entre a entrada e saída. Para usar uma função, você precisa defini-la em algum lugar no escopo a partir do qual deseja chamá-la.

Uma definição de função (também chamada de declaração de função ou instrução de função) consiste na palavra-chave da função, seguida por:

- O nome da função.
- Uma lista de parâmetros para a função, colocados entre parênteses e separados por vírgulas.
- As instruções JavaScript que definem a função, entre colchetes.

(MDN Web Doc , 2017)

No estudo de caso referido, será necessário criar uma série de funções para atingir nosso objetivo.

Lembrando que o objetivo é variar tamanho e cor, será necessário criar pelo menos as funções `fill ()` e `rect ()` com suas variáveis internas.

Antes de tudo será necessário declarar as variáveis inserindo seus valores iniciais em setup ()

```
let viz1, viz2, viz3, viz4, viz5;
function setup() { viz1 = 50;
viz2 = 40;
viz3 = 40;
viz4 = 30;
viz5 = 60;
}
```

Variáveis declaradas com let terão um escopo de bloco. Isso significa que a variável existe apenas dentro do bloco em que foi criada.

Neste ponto, é necessário criar uma função sectionViz () baseada nas variáveis (x, y, viz), configurando a função retângulo de acordo com estas variáveis:

```
function sectionViz (x, y, viz) {

let l = 100;
let a = 100;

rect (x, y+(a-viz), l, viz, 5);

}
```

Finalmente, na função draw () vamos inserir os retângulos com base na função sectionViz e nas variáveis globais:

```
Function draw() {
sectionViz (width/2-50, height/5, viz1);
sectionViz (width/4-48, height/5*2, viz5);
sectionViz (width/4*3-50, height/5*2, viz4);
sectionViz (width/4-48, height/5*3.5, viz2);
sectionViz (width/4*3-50, height/5*3.5, viz3);

}
```

finalmente, para criar a interação do teclado, usaremos o evento `keyPressed ()`. O evento `keyPressed ()` é chamado uma vez sempre que uma tecla é pressionada. O `keyCode` da tecla pressionada é armazenado na variável `keyCode`. Para facilitar, usaremos os `keyCode` `DOWN_ARROW` e `UP_ARROW`.

Vamos lembrar o que gostaríamos de dizer com esta linguagem: *“se o número de animais varia positivamente, então o retângulo interno cresce e fica cada vez mais verde. Se o número de animais varia negativamente, o retângulo interno diminui e fica cada vez mais vermelho ”*

Além disso,

“Se o número de animais varia positivamente, o fator X varia positivamente e o retângulo dentro dele cresce e fica cada vez mais verde. Se o número de animais varia negativamente, o retângulo dentro dele diminui e fica cada vez mais vermelho ”.

Tendo 5 fatores diferentes (`viz1`, `viz2`, `viz3`, `viz4`, `viz5`), gostaríamos da seguinte correlação entre eles:

Se `viz1` aumenta, `viz3` e `viz5` aumentam e `viz2` e `viz4` diminuem.

Inversamente, se `viz1` diminui, então `viz3` e `viz5` diminuem e `viz2` e `viz4` aumentam.

Cada fator pode ter um valor mínimo igual a 0 e um valor máximo igual a 100 para ficar dentro de um quadrado.

Portanto, quando a ação `keyPressed ()` com `keyCode` `DOWN_ARROW` estiver ativa e `viz1 > 0`, `viz2 < 100`, `viz3 < 0` e `viz4 < 0`, `viz1` diminuirá, `viz2` aumentará, `viz3` diminuirá, `viz4` aumentará e `viz5` diminuirá. Nos códigos, isso é representado da seguinte maneira:

```
function keyPressed() {  
  if (keyCode===DOWN_ARROW && viz1>0 && viz2<100 && viz3>0 &&  
    viz4<100) {  
  
    viz1--;  
    viz2++;  
    viz3--;
```

```
viz4++;  
viz5--;  
  
}
```

Quando a ação keyPressed () com keyCode UP_ARROW estiver ativa e viz1 <100, viz2 > 0, viz3 <100 e viz4 > 0, viz1 aumentará, viz2 diminuirá, viz3 aumentará, viz4 diminuirá e viz5 aumentará. Nos códigos, isso é representado da seguinte maneira:

```
if( keyCode===UP_ARROW && viz1<100 && viz2>0 && viz3<100 &&  
viz4>0){  
  
viz1++;  
viz2--;  
viz3++;  
viz4--;  
viz5++;  
  
}
```

Neste ponto, para criar as mudanças de cor, será necessário redefinir o modelo para exibir a cor em HSB usando a função:

```
colorMode (HSB, 360, 100, 100, 100);
```

HSB é um dos menos conhecidos, mas certamente não menos importantes métodos de composição de cores, sigla para Hue Saturation Brightness, mas também é chamado de HSV, Hue Saturation Value (value), ou HSI, Hue Saturation Intensity (intensidade). Como já referido, é um método aditivo para a composição de cores e sua representação em sistema digital e está particularmente orientado para a perspectiva humana, baseando-se na percepção que se tem de uma cor em termos de tonalidade, tonalidade e tom.

O modelo HSB é um sistema de representação de cores por meio de 3 valores:

1. **H (Hue): Matiz.** É expresso em graus de 0 a 360 °. É o que comumente chamamos de cor, então é vermelho, amarelo, verde etc ... independentemente de serem claros ou escuros, claros ou opacos, intensos ou cinza.
2. **S (Saturation): Saturação.** Possui valor percentual de 0 a 100%. Representa o grau de intensidade de uma cor, na experiência visual, quanto mais se torna cinza menos saturada fica.
3. **B (Brightness): Brilho.** Possui valor percentual de 0 a 100%. Representa o grau de brilho de uma cor, ou seja, a quantidade de luz da cor, que expressamos ao percebê-la como “clara” ou “escura”.

Exceto para os daltônicos, a compreensão do conceito de tonalidade é imediata. Vamos pegar uma imagem (ver fig. 13) feita de 3 quadrados com H (matiz) a uma distância de 60 ° e com saturação e brilho sempre a 100% (então não precisamos nos preocupar com isso). Vamos observar como o matiz varia conforme o valor de H.

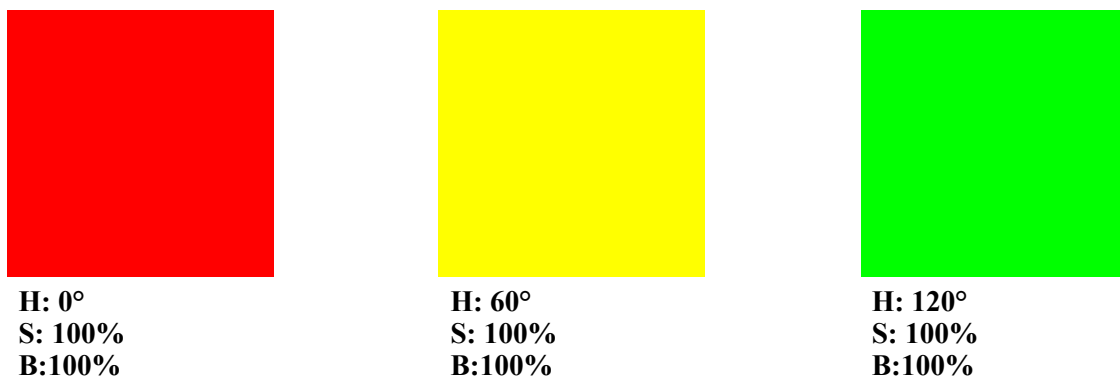


Fig. 13 Cores no modelo HSB

Na verdade, é precisamente sobre o valor de H que vamos trabalhar criando uma função fill () composta por 3 variáveis internas: *h*, *s* e *b*.

A variável *h* terá uma função map () dentro dela, ou seja, ela mapeia novamente um número de um intervalo para outro de acordo com uma função map(value, start1, stop1, start2, stop2, [withinBounds]).

O parâmetro *value* será representado pela variável global viz, enquanto os parâmetros start1 será igual a 0 e stop1 será igual a 100 e representam os parâmetros mínimo e máximo em relação a viz em que a variável muda, start2 será igual a 0 e stop2 será igual a 120 e representam os dois parâmetros mínimos e máximos relativos ao valor de h, ou seja, à percentagem de tons de cor. A função de mapa é uma função geral que pega qualquer intervalo e mapeia um valor dentro desse intervalo para um novo valor em qualquer outro intervalo.

Quanto mais a variável viz aumenta, mais verde será a cor de preenchimento. Quanto mais a variável viz diminui, mais a cor do preenchimento será mais e mais vermelha.

Na linguagem de programação, isso se traduzirá em:

```
Function sectionViz(x,y,viz) {  
  Let h= map(viz,0,100,0,120);  
  Let s = 100;  
  Let b = 100;  
  fill(h,s,b,);  
}
```

Relembrando a *conditional statement*:

```
function keyPressed() {  
  if (keyCode===DOWN_ARROW && viz1>0 && viz2<100 && viz3>0 &&  
  viz4<100) {  
  
    viz1--;  
    viz2++;  
    viz3--;  
    viz4++;  
    viz5--;  
  
  }
```

```

if(  keyCode===UP_ARROW  &&  viz1<100  &&  viz2>0  &&  viz3<100  &&
viz4>0){

viz1++;
viz2--;
viz3++;
viz4--;
viz5++;

}

```

Em sua totalidade, esta parte do protótipo se traduzirá em:

```

let viz1, viz2, viz3, viz4, viz5;
function setup() {
viz1 = 50;
viz2 = 40;
viz3 = 40;

viz4 = 30;

viz5 = 60; }

function sectionViz (x, y, viz) { let l = 100;
let a = 100;

    colorMode (HSB, 360, 100, 100, 100);
let h = map(viz, 0, 100, 0, 120); let s = 100;
let b = 100;

    fill (h, s, b);
rect (x, y+(a-viz), l, viz, 5);

```

```

Function draw(){
colorMode (HSB, 360, 100, 100, 100);}

    fill(216,58,80);
    rect(width/2-50,height/5,100,100,10);
    fill(328,98,94);
    rect(u,e,c,t,10);
    fill(162,47,87);
    rect(r,e,c,t,10);
    fill(190,100,77);
    rect(u,s,c,t,10);
    fill(208,93,64);
    rect(r,s,c,t,10);
    sectionViz (width/2-50, height/5, viz1);
    sectionViz (width/4-48, height/5*2, viz5);
    sectionViz (width/4*3-50, height/5*2, viz4);
    sectionViz (width/4-48, height/5*3.5, viz2);
    sectionViz (width/4*3-50, height/5*3.5, viz3);
function keyPressed() {
if (keyCode===DOWN_ARROW && viz1>0 && viz2<100 && viz3>0 &&
viz4<100) {
    viz1--;
    viz2++;
    viz3--;
    viz4++;
    viz5--;
}
if( keyCode===UP_ARROW && viz1<100 && viz2>0 && viz3<100 &&
viz4>0){
    viz1++;

```

```

viz2--;
viz3++;
viz4--;
viz5++; }

```

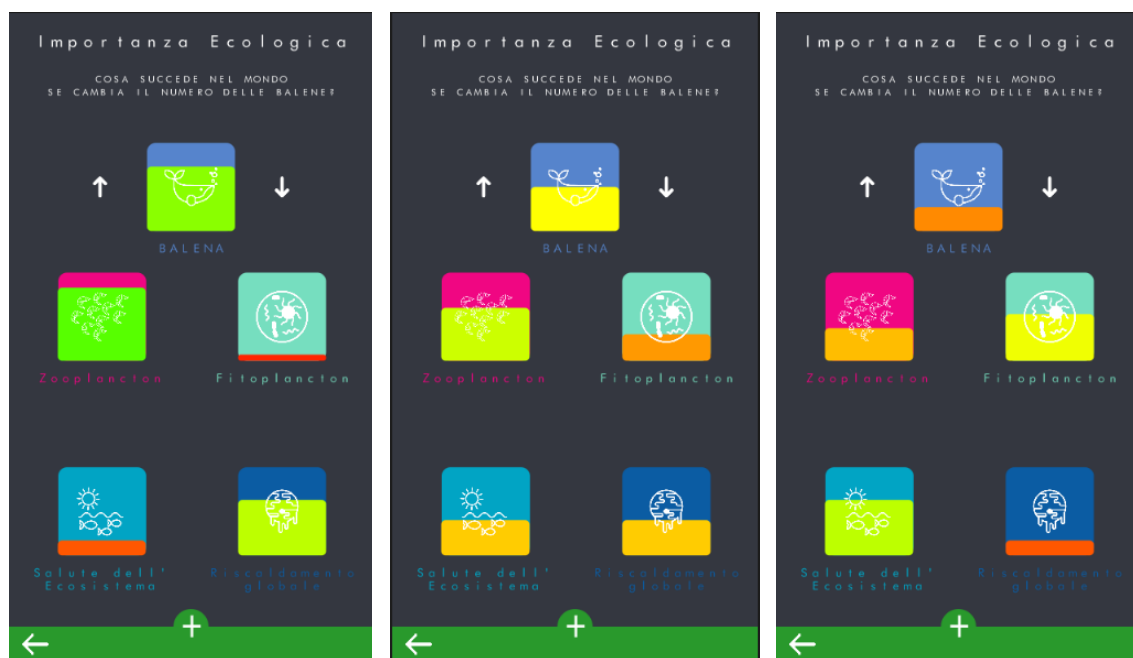


Fig. 14. Exemplo de mudança de cor dependendo do modelo descrito criadas durante o projeto desenvolvido durante o mestrado em design de interação

Para tornar o protótipo mais intuitivo e envolvente, será preciso inserir imagens, vídeos, entradas e textos. Vejamos abaixo como criar interações e animações graças a esta ferramenta de codificação.

Para inserir um vídeo, iremos inserir uma função `preLoad()`.

Chamada diretamente antes de `setup ()`, a função `preload ()` é usada para lidar com o carregamento assíncrono de arquivos externos de forma bloqueadora. Se uma função de pré-busca for definida, `setup ()` irá esperar até que as chamadas de upload da extensão sejam completadas. Nada além das chamadas de carregamento (`loadImage`, `loadJSON`, `loadFont`, `loadStrings`, etc.) deve estar dentro da função `preload`. Se o carregamento assíncrono for preferido, os métodos de carregamento podem ser chamados em `setup ()` ou em qualquer outro lugar com o uso de um parâmetro de retorno de chamada.

A função `preload` define a variável relativa a um objeto, contida na pasta de imagens de p5.

Portanto, primeiro você precisa inserir a imagem na pasta relativa às imagens de p5.

A seguir, declararemos a variável `vid`, relativa ao vídeo que queremos inserir.

```
var vid;
```

Existem três tipos de declarações em JavaScript:

1. **var** Declaração de uma variável, opcionalmente inicializada com um valor.
2. **let** Declaração de uma variável local com visibilidade no bloco, opcionalmente inicializada com um valor.
3. **const** Declaração de uma constante somente leitura com um nome.

Variáveis são nomes simbólicos a serem usados em aplicativos no lugar dos valores que representam. Nomes de variáveis, chamados de identificadores, devem seguir certas regras de escrita. Um identificador JavaScript deve começar com uma letra, sublinhado (`_`) ou cifrão (`$`), enquanto os caracteres subsequentes também podem ser números (0-9). Como o JavaScript diferencia maiúsculas de minúsculas, as letras incluem os caracteres "A" a "Z" (maiúsculas) e os caracteres "a" a "z" (minúsculas).

(MDN Web Doc [moz://a](https://developer.mozilla.org/en-US/docs/Glossary/Identifier), 2017)

Quando uma variável é declarada fora de qualquer função, ela é chamada de variável global, pois está disponível em todo o código do documento atual. No entanto, quando a variável é declarada em uma função, ela é chamada de variável local, pois só está disponível nessa função.

Nesse caso, a variável `var` é uma variável local e tem valor apenas dentro da função em que é declarada. Na verdade, a variável é destruída automaticamente no final da função. Variáveis locais são aquelas que são declaradas nos subprogramas individuais, portanto, são válidas (visíveis) apenas dentro deles e, portanto, não podem ser usadas em outros subprogramas. Portanto, eles têm uma duração, nascem com a função (ou sub-rotina) e morrem quando a própria função termina. (Coding Criativo, 2020)

As declarações de variáveis, onde quer que apareçam, são processadas antes que qualquer código seja executado. O ambiente de uma variável declarada com `var` é seu contexto de execução atual, que é a função de fechamento ou, para variáveis declaradas fora de qualquer função, global.

Assim que a variável for declarada, ela precisará ser definida. Como este é um objeto, ele será definido em uma função `preload()` antes da função `setup()`. Em seguida, criaremos a função `createVideo` como uma declaração do que está para acontecer. Dentro da variável `vidload` que definiremos imediatamente após o `preload()`.

```
function preload(){  
  vid = createVideo("images/giphy.mp4", vidLoad);  
}
```

A variável `vid` representa como o vídeo é carregado. Essa variável é posteriormente definida, na função `vidLoad()`, definindo com precisão os métodos de carregamento:

```
function vidLoad(){  
  vid.hide();  
  vid.loop();  
}
```

vid.loop é um atributo que faz com que o vídeo comece novamente, sempre que termina.

Por fim, vamos usar o vídeo na função draw ().

```
function draw() {  
  imageMode(CENTER);  
  image(vid, width/2, height/2, 1308, 736);  
}
```

Tudo isso que dissemos até agora se traduz na seguinte linguagem:

```
var vid;  
function preload() {  
  vid = createVideo("images/giphy.mp4", vidLoad);  
}  
  
function vidLoad() {  
  vid.hide();  
  id.loop();  
}  
function draw() {  
  imageMode(CENTER);  
  image(vid, width/2, height/2, 1308, 736);  
}
```

Outro tipo de interação necessária no caso de um aplicativo é o input. A função createInput () é chamada quando um input é encontrado. Neste caso, cada vez que input1 é ativado, o criador de entrada é ativado.

```
let input1;  
  
function preload() { input1= createInput();  
  
  input1.hide();
```



```

}
function draw(){
fill(0,0,0,0);
input1.show();
input1.size(180,40);
input1.position(width/2-95,height/4*3-75);

}

```

Os efeitos sonoros fornecem profundidade a todos os gráficos do aplicativo, dão vida à cena, estabelecem o clima do aplicativo e fornecem feedback útil para o utilizador. Os sons devem ser consistentes com a ação à qual estão ligados ou o significado que pretendem expressar e seu volume geral deve ser moderado. No entanto, o aplicativo deve permitir que você gerencie esses volumes e, se necessário, excluir completamente os efeitos sonoros.

O uso de p5.js exigirá a inclusão da biblioteca p5.sound. Você precisa colocar o seguinte no início do arquivo index.html:

```
<script src = "path / to / p5.sound.js"> </script>
```

O objeto SoundFile comporta-se como um arquivo normal, como uma imagem ou um vídeo, portanto, será necessário:

1. Defina a variável mysound.
2. Inserir o comando na função de pré-carregamento ().
3. Criar um evento de acordo com o qual o som é ativado

```

Var mySound;
function preload() {
soundFormats('mp3', 'ogg');
mySound = loadSound('assets/BlueWhale');
}

```

```
function mousePressed()    if ( mouseX > 0 && mouseX < width &&  
mouseY > 0 && mouseY < height-50){  
mySound.play();  
  
}
```

Existem muitos tipos de animações e interações que podem ser criadas. Ao projetar um aplicativo para crianças, é preciso sempre concentrar-se na simplicidade. As operações mais intuitivas possível ajudarão na navegação e a experiência da criança será mais agradável.

Como designer, compreender as funcionalidades e os fundamentos da linguagem java script permite que se tenha uma visão mais ampla do processo de criação do produto.

6. Conclusões

O presente trabalho trata da identificação de uma solução digital adequada para o ensino da educação ambiental nas escolas.

Neste estudo analisamos as várias fases de um processo de criação de uma aplicação, desde a definição de um problema até à criação de uma possível solução de acordo com o objetivo a que se refere.

Acima de tudo há a idéia de que as crianças estão na base do futuro da nossa sociedade e, para melhorar os múltiplos aspectos disso, é necessário dar às crianças todos os meios para possam aprender, conhecer e estudar todos os aspectos do nosso mundo. A tecnologia é o meio que nos permite fazer isso sem comprometer a qualidade e a quantidade de informações.

Enriquecer a experiência de ensino com as tecnologias é uma necessidade inevitável de uma escola que pretenda enfrentar e gerir os desafios de uma sociedade cada vez mais moldada pelos ritmos e regras da rede.

No entanto, este estudo pretende ser um guia para visualizar as etapas de implementação de um produto digital de e-learning, como o de um aplicativo como ferramenta de ensino de educação ambiental.

As novas tecnologias de aprendizagem criam novas oportunidades para inovar a sala de aula tradicional, tornando-a mais envolvente e dinâmica. É preciso saber aceitar as mudanças de forma positiva e saber mudar a atitude em relação às tecnologias e à aprendizagem.

Ainda há um longo caminho a percorrer e podem existir diferentes soluções que permitirão a criação de experiências envolventes e altamente personalizadas.

Nesta perspectiva, será sempre fundamental a colaboração com várias figuras profissionais e várias equipas de trabalho que colaboram entre si num processo de criação.

Este estudo complementa a literatura existente sobre novas tecnologias e estudos relacionados ao Child Center Design com as avaliações feitas durante essa experiência.

Antes de poder pensar em projetar aplicativos para crianças, é necessário estudar a fundo seu comportamento diante da tecnologia, pois eles têm um uso muito diferente do dos adultos.

Neste estudo analisamos as várias etapas do processo de criação de um design para crianças deste tipo, analisando também duas metodologias de processo de design: Processo de Design Thinking e Processo de Design de Produto. No entanto, acreditamos que ambos os métodos são válidos por se basearem num ritmo não linear do processo. Aliás, a não linearidade do processo baseado em pesquisas e testes contínuos em cada etapa permite a criação de um produto cada vez mais semelhante e adequado para o utilizador, identificando problemas críticos e podendo corrigi-los nas várias fases.

A fase central do processo de criação de um design deste tipo é a prototipagem precedida de um estudo do que deve ser a linguagem visual utilizada.

Como mencionado várias vezes, criar um design para crianças é muito mais do que tornar as regras de criação de um design adulto mais coloridas e vibrantes. As crianças têm suas próprias necessidades e exigências que mudam a cada estágio de seu crescimento. Tendo avaliado um target entre 10 e 12 anos, acreditamos ser fundamental criar um design que não seja muito infantil, mas seja simples no uso, usando cores fortes, fontes legíveis, tamanhos de botões adequados para uso fácil, iconografia que permita clareza, legibilidade dos comandos, uso de feedback auditivo, uso de pelo menos um personagem narrativo e que as ações são projetadas de acordo com o uso por meio de gestos intuitivos como tocar, arrastar e deslizar.

Uma vez que uma interface gráfica tenha sido planeada, será útil prosseguir com testes de usabilidade não moderados para entender as preferências gráficas e de uso das crianças, mostrando-lhes perguntas claras e simples. Já com testes de usabilidade moderados será possível, graças à comparação direta, entender melhor suas expectativas.

Proceder várias vezes e nas várias fases de prototipagem com estes tipos de testes de usabilidade será essencial para perceber se estamos a proceder da forma certa, compreendendo as várias críticas.

Por fim, neste estudo, analisamos os vários tipos de prototipagem avaliando a utilização dos designers do programa p5.js que lhes permite aproximarem-se do trabalho dos técnicos de programação em particular, permitindo um melhor entendimento entre os dois profissionais.

Infelizmente, os estudos relativos à faixa etária por nós estudada são relativamente poucos e é fundamental observar as crianças diretamente.

Esta tese pretende ser um guia dos procedimentos necessários para poder criar aplicações futuras para crianças cada vez mais focadas na sua forma de utilização deste tipo de tecnologia.

As crianças são as que podem ter maiores oportunidades de crescimento e conhecimento graças à tecnologia, mas isso só pode acontecer se nós, adultos, formos capazes de criar ferramentas sob medida para elas.

7. Bibliografia

Saffer D. (2007) Design dell'interazione, creare applicazioni intelligenti e dispositivi ingegnosi con l'interaction design. Milano: Paravia Bruno Mondadori Editori

Moggridge B. (2006) *Designing Interactions*. Cambridge: The MIT Press

Thackara J. (2005) In the Bubble, Designing in a Complex World. Cambridge: The MIT Press

Cecchi A. & Belloni V. (2009) Processing. Guida introduttiva alla programmazione visuale.

Rubin J. & Chisnell D. (2008) Handbook of Usability Testing, John Wiley & Sons

Visciola M. (2006) L'usabilità dei siti web: curare l'esperienza d'uso in internet, Apogeo

Baxter K., Courage C. & Caine K. (2005) Understanding your users, Morgan Kaufmann Publishers

Hartson, Rex & Pyla, Partha S. (2012) The UX Book Morgan Kaufmann Publishers

Gelman D. L. (2014) Design for Kids, digital product for playing and learning, Rosenfeld Media Brooklyn

Cuello J. & Vittone J. (2013) Designing Mobile App, CreateSpace Independent Publishing Platform

Dumas J. S. & Loring B. A. (2008) Moderating Usability Tests, Principles and Practices for Interacting, Morgan Kaufmann Publishers

Nuez A. (Retrieved September 10 2009) An Attainable Goal: Quantifying Usability and User Experience. *User Experience Magazine*, Volume 7, Issue 3

Farnsworth C. (Retrieved September 10 2009) Getting Your Money Back: The ROI of Remote Unmoderated User Research. *User Experience Magazine*, Volume 7, Issue 3

Reas C. & Fry B. (2010) Getting started with Processing, Maker Media

Horton I. (2002) Java 2 SDK 1.4. Guida per il programmatore, Wrox Pr Inc

Flanagan D. (2002). Javascript. Versione 1.5. La guida, Apogeo

Greenberg I.(2006). Processing - Creative Coding and Computational Art , Apress

Noble J. (2009). Programming Interactive - A Designer's Guide to Processing, Arduino, and openFrameworks, O'Reilly Media

Badger M.(2014). Scratch 2.0 Beginner's Guide - Create digital stories, games, art, and animations through six unique projects, Packt Publishing

Dumas J. S. & Loring B. A. (2008) Moderating Usability Tests, Principles and Practices for Interacting, Morgan Kaufmann

Costa B., Franco T. e Grilo M. (2019) Product Design Process: The manual for Digital Product Design and Product Management, Imaginary Cloud Limited []

Soucy K. (2010) INTERACTION DESIGN FOUNDATION (2020) The Basics of User Experience Design Unmoderated, Remote Usability Testing: Good or Evil?

Newman D. (2018) Top 5 Digital Transformation Trends In Education For 2019, Forbes [<https://www.forbes.com/sites/danielnewman/2018/11/13/top-5-digital-transformation-trends-in-education-for-2019/#6274e6085d4ddanielne>]

Menegazzi D., Sylla C. & Padovani S. (2018) A Preliminary Study of Interactivity on Visual Narrative in Children's Story Apps, Research Gate

Parker & Quin (2019). Technology in Education 2019: Trends to Watch. . Acesso em 25 de Maio de 2019. []

Hamilton L. C.(2011) Education, politics and opinions about climate change evidence for interaction effects, *Climatic Change* **104**, 231–242 (2011). []

Software libero & TD, Primi passi con p5.js (2018) []

Computer Hope, Conditional Statement (2017) []

Personas []

Coding Creativo (2020) []

MDN Web Doc [moz://a](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array), (2017) []

