

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E
COMPUTAÇÃO CIENTÍFICA



MS777: Projeto Supervisionado I

Sobre um aplicativo para gerenciar filas de dúvidas em iOS

Aluno: LUCAS CAVALCANTE DE BARROS - RA117706
LucasCavalcanteDeBarros@gmail.com

Orientadora: PROF. DR. SANDRA AUGUSTA SANTOS
Sandra@ime.unicamp.br

RESUMO

Esse trabalho aborda a trajetória de desenvolvimento do aplicativo FilaFácil usado no auxílio de atendimento de dúvidas dos alunos, nas plataformas iOS, TvOS e watchOS. O texto tem como intuito mostrar as etapas do desenvolvimento de um *software* desde a concepção até a publicação na *AppStore*, além de manutenção e atualização do mesmo devido a *bugs* e *feedback* dos usuários. Mostra a perspectiva de desenvolver um programa sozinho tanto como em uma equipe formada por pessoas especializadas em áreas diferentes, focando em dificuldades e soluções durante o processo de desenvolvimento.

ABSTRACT

This report describes the trajectory of the developing of FilaFácil, used to assist in the attending of students doubts, in the iOS, TvOS and watchOS platforms. The text intends to show the steps of developing a software from the idealization till the publishing in the AppStore, besides maintaining and updating due to user's feedback and bugs found during testing. It shows a perspective of developing a program by oneself and also in a team of people, each specialized on different areas, focusing the narrative in the difficulties and solutions found during the developing process.

Sumário

DESCRIÇÃO DO PRODUTO	4
OBJETIVOS & QUESTIONAMENTOS INICIAIS	4
DESENVOLVIMENTO	5
FASE 1: Desenvolvimento individual	5
FASE 2: Desenvolvimento em equipe para AppleTV	5
FASE 3: Desenvolvimento em equipe para Apple Watch	5
FASE 4: Submissão para AppStore	5
RESULTADO	6
TRABALHO FUTURO	6
CONSIDERAÇÕES FINAIS	6
REFERÊNCIAS BIBLIOGRAFICAS	7
WEBGRAFIA	7

1. DESCRIÇÃO DO PRODUTO

FilaFácil é um aplicativo para administrar filas de dúvidas de alunos, baseado em áreas de conhecimento específicas, voltadas para um curso particular o qual tem as disciplinas de *Design*, Desenvolvimento e Empreendedorismo. Primeiramente, o usuário deve entrar na sua conta, acessando a tela que mostra uma lista de dúvidas de seus colegas por ordem de chegada e filtrada por área de conhecimento.

Após estar logado, o aluno pode criar uma nova dúvida, escrevendo a pergunta e escolhendo uma das áreas. Assim que criada, ele entra no final da fila estando disponível para todos seus colegas verem e poderem procurá-lo para sanar sua dúvida, além do professor responsável por aquela área receber uma notificação no seu celular avisando que um aluno entrou na fila.

Tanto o aluno como o professor também podem acessar a aba de avisos e escrever um aviso geral para toda turma, como sala de prova, aula cancelada e *site* oficial da disciplina. Todas essas informações, como dúvidas e avisos, estão sempre visíveis para a turma na aplicação de TV, para facilitar a transparência da fila e disponibilizar as dúvidas para que os usuários também possam sanar as dúvidas de colegas.

2. OBJETIVOS & QUESTIONAMENTOS INICIAIS

Objetivo principal: Intermediar a comunicação entre professores e alunos, garantindo um atendimento organizado e por ordem de chegada, além de oferecer visibilidade para alunos se ajudarem, aumentando a interação da comunidade e reduzindo a carga sobre os professores.

Questão Principal: Como o aplicativo poderia ajudar tanto professores quanto alunos, de forma organizada, simples e transparente?

Questões importantes para orientar o projeto:

- Quais as dificuldades para adesão do produto pelos professores?
- Quais as dificuldades para adesão do produtos pelo alunos?
- Quais as dificuldades dos professores sentem para perceber dúvidas?
- Quais as maiores dificuldades enfrentadas pelos alunos para tirar dúvidas?
- Como resolver o problema mantendo a solução o mais simples possível?
- Como facilitar o uso para professores mais conservadores?
- Como incentivar os usuários a continuar usando a solução mesmo depois de tirar a dúvida?

3. DESENVOLVIMENTO

O desenvolvimento do projeto pode ser dividido em duas etapas principais: O desenvolvimento individual, quando fiz como projeto pessoal, e quando me juntei com um grupo de colegas e montamos uma equipe para terminar o desenvolvimento do projeto. A seguir descrevo as principais etapas em que ocorreram as mais significativas mudanças no projeto.

3.1. FASE 1: Desenvolvimento individual

Motivação: No curso particular em que estudamos, somos incentivados a desenvolver aplicativos que resolvam um problema. Desde que cursei Licenciatura em Física na Unicamp sempre tive interesse na área de ensino, cursando disciplinas de Ensino de Matemática para entender melhor técnicas de ensino e dificuldades de alunos. Durante as aulas eram muito incentivadas a participação e a troca de conhecimento entre os próprios alunos. Da mesma forma acontecia no curso no curso particular de desenvolvimento que estou cursando, em que os alunos devem buscar o conhecimento com a ajuda dos colegas e professores, logo queria desenvolver algo que ajudasse nesse processo.

Em fevereiro de 2018, ao observar o quadro de atendimento de uma das professoras do curso, tive dificuldade em saber qual seria a ordem correta, já que era baseado em *Post-its* e cada alunos os colava de forma diferente, ou as vezes perdiam a cola e caiam no chão. Percebendo essa necessidade da professora de organizar a fila de atendimento, além de ter que sempre recorrer ao quadro para saber se tinha alguém com dúvidas, pensei que poderia resolver o problema fazendo um aplicativo com uma fila em formato de lista, onde os alunos poderiam acrescentar seus nomes com a dúvida.

Prototipação: Tentando manter o projeto o mais simples possível, para ser desenvolvido rapidamente e validar se a idéia tinha valor para professora, optei por um *design* visual simples baseado nas cores verde e preto para criar uma identidade do aplicativo. Já que o curso tinha uma ferramenta ao qual as cores base eram tons de laranja, pensei em manter um padrão para que assim cada ferramenta tivesse uma cor própria e fosse intuitivo para os alunos identificarem. Primeiramente fiz apenas uma lista em que podiam ser adicionados novos itens sem armazenar informações em lugar nenhum, e mostrei para a professora, que gostou da ideia e me incentivou para que eu desenvolvesse a aplicação.

Nas semanas seguintes criei um banco de dados em *Firebase* [14] para criar e salvar informações de usuários, além de armazenar dúvidas no formato Json, por ser um banco *NoSQL*¹ [33, 34]. Optei por esse banco por ser facilmente portável para Android, caso seja necessário futuramente, e por já possuir experiência trabalhando com esse banco de dados, agilizando assim o processo de desenvolvimento. Em relação ao aplicativo, escolhi desenvolver usando arquitetura *MVC*²[35, 36] por ser a mais simples e possuir um nível de desacoplamento entre os elementos do aplicativo, sendo assim possível futuramente substituir partes do projeto sem gerar grandes alterações e retrabalho.

Outra importante ferramenta que usei durante o desenvolvimento foi o *SwiftLint*, que tem o propósito de manter a qualidade do código. Essa ferramenta gera avisos leves ou graves, dependendo da regra quebrada, sendo possível criar regras de qualidade de código, como número máximo de caracteres na mesma linha. O intuito era manter o código limpo e comprehensível em caso de manutenção ou entrada de novos membros no projeto, como irei descrever mais adiante.

¹ NoSQL: Banco de dados que armazena e recupera dados de forma diferente de bancos de dados relacionais

² MVC: Model-View-Controller é um padrão de arquitetura de software que separa a representação da informação da interação do usuário com ele

Descrição do Protótipo:

Na imagem 1 abaixo: da esquerda para direita, a primeira é a tela de início com nome do aplicativo e o logotipo inicial que desenhei. A segunda é a tela da fila de dúvidas dos alunos, apresentando o nome do aluno acima do texto da dúvida, assim como uma foto do perfil e a sua colocação na fila. Na aba superior há um filtro com todos alunos com dúvida ou selecionar por algum professor em específico como "Guga" ou "Sergio". Já a terceira, é a tela onde o aluno pode abrir uma dúvida, escrevendo a pergunta no primeiro campo e selecionando qual professor gostaria que sanasse sua dúvida, e no final clicar no botão verde para entrar na fila.

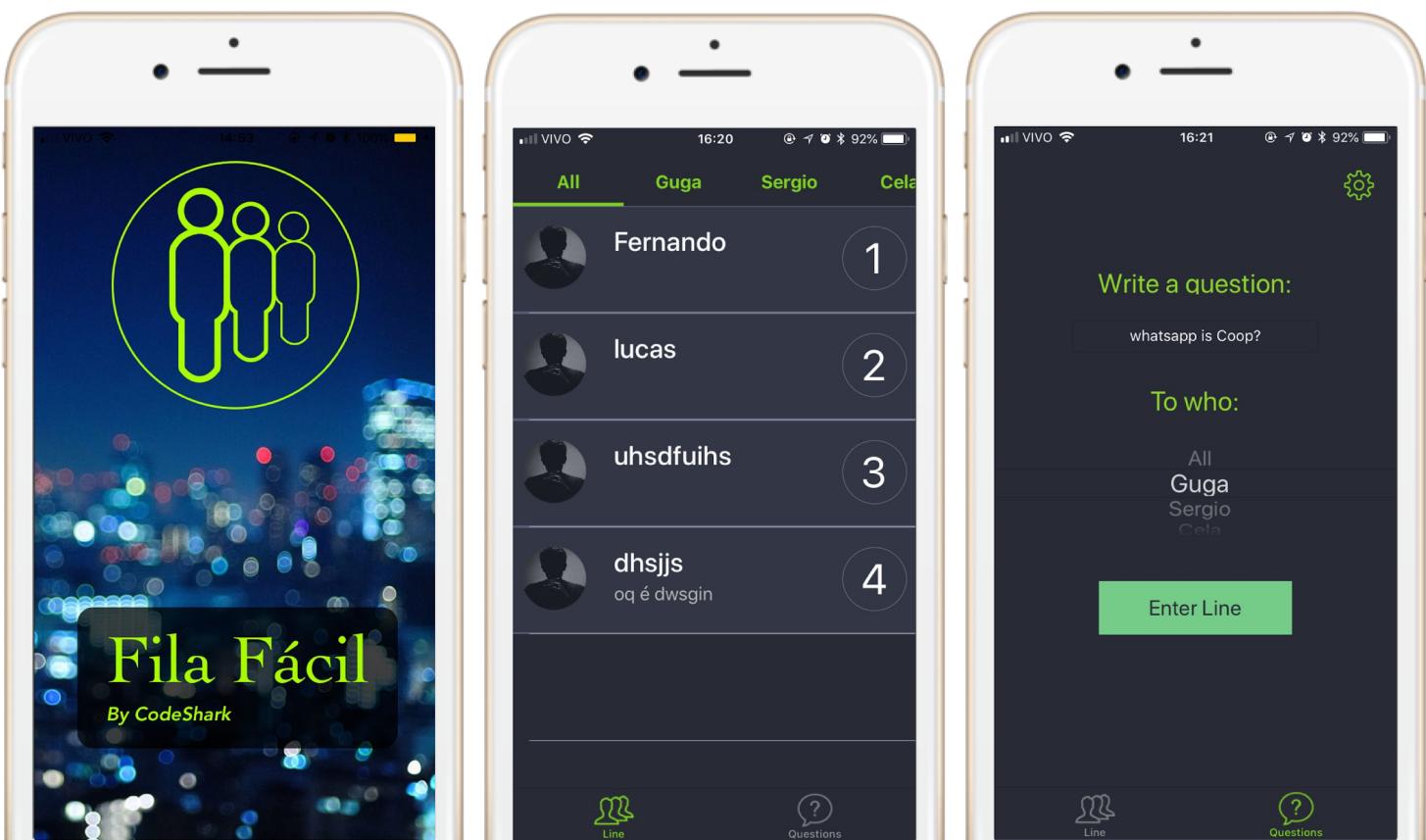


Imagen 1: Da esquerda á direita, Tela de entrada, Tela de fila e Tela de criar dúvida

Na imagem 2 abaixo: da esquerda para direita: A primeira tela contém a visão do celular do professor, mostrando quem é o próximo aluno na fila, há quanto tempo esse aluno já está esperando e quantos alunos estão na fila no total, assim como o botão verde para indicar que o aluno foi atendido e chamar, para ver quem é o próximo, sem ter que consultar a lista da fila. A segunda tela mostra o perfil do usuário, indicando seu nome, e-mail e tipo de perfil, se é aluno ou professor. Já a última tela mostra a visão no celular do aluno, indicando qual sua colocação na fila e qual pergunta ele fez, além do botão vermelho, caso ele queira sair da fila por já ter sanado sua dúvida.

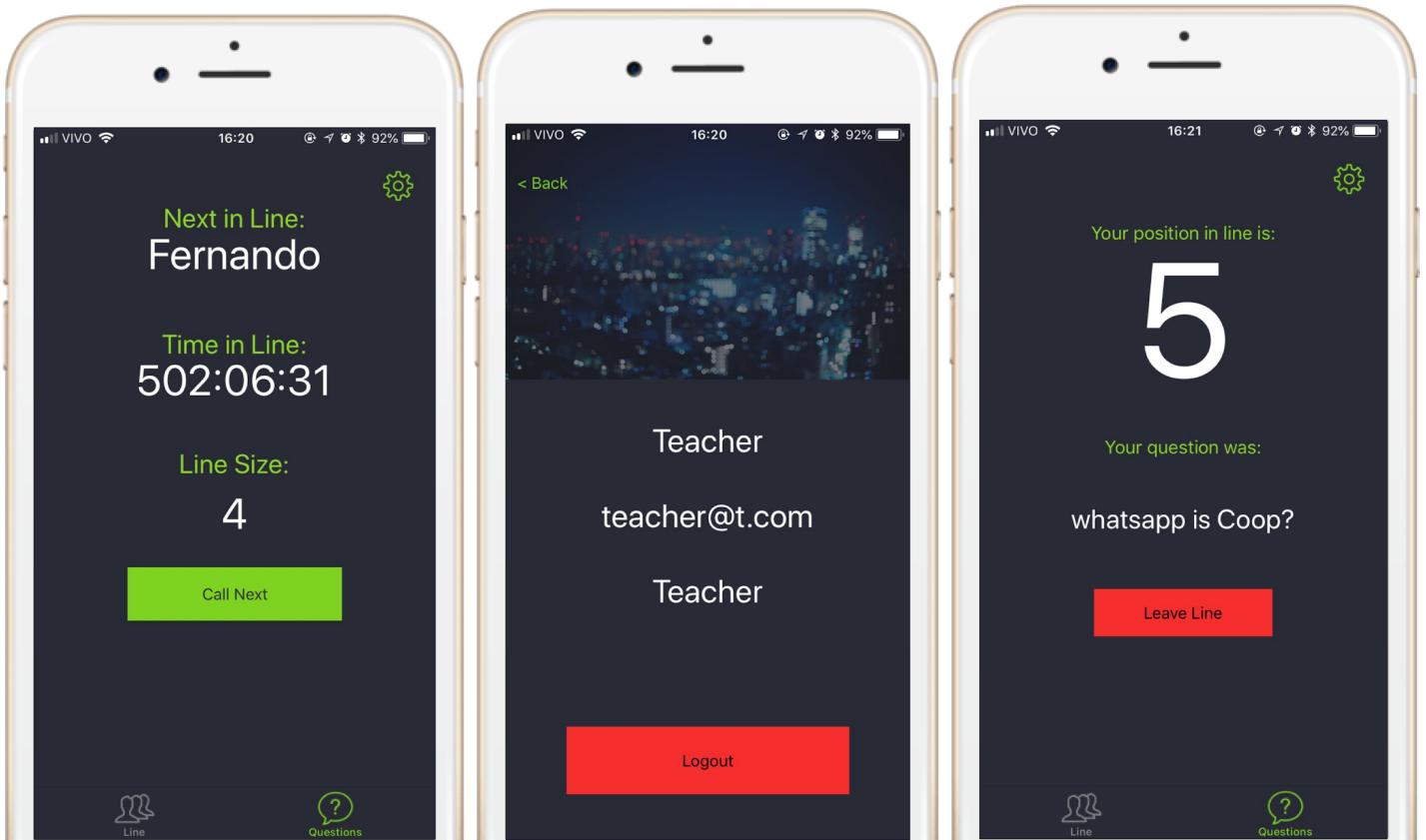


Imagen 2: Da esquerda á direita, Tela de acompanhar colocação na fila para aluno, Tela de perfil e Tela de status da fila para professor

Validação: Durante o desenvolvimento do protótipo mais sofisticado com *login*, senha e banco de dados, fui mostrando o resultado para os professores do curso e coletando *feedbacks* do que poderia corrigir ou melhorar até chegar em um MVP (Produto Mínimo Viável) de acordo com a definição de F. Robinson [29]. Essa etapa foi extremamente importante pois trabalhar próximo aos professores ajudava a dar prioridade a funcionalidades e evitar erros ou retrabalho. Pude ainda aprender a trabalhar tanto com críticas positivas da professora que achou uma boa ideia e pediu para avisar quando estivesse pronto para ela divulgar para colegas, quanto de outro professor, que achou que o aplicativo não seria tão interessante por ele estar sempre perto dos alunos caso tivessem

dúvida.

Dificuldades e Aprendizados: Apesar de ter trabalhado com *Firebase* em projetos passados, tive bastante dificuldade em manter o banco de dados atualizado em tempo real com o aplicativo, já que havia consultas assíncronas ao banco que demoravam mais tempo do que o tempo para renderizar a informação na tela do usuário, ficando muitas vezes desatualizada ou com dados errados. Essa questão foi uma das minhas maiores dificuldades, levando entre 2 a 3 semanas para ser solucionada. Um hábito que criei e achei muito útil para futuras referências é de documentar todos os sites e tutoriais que usei ao longo do desenvolvimento, caso fosse necessário saber como fazer algo novamente ou verificar a fonte da informação. Fiz uso principalmente de artigos do *StackOverFlow*[38], famosa comunidade de desenvolvedores que compartilha dificuldades, conhecimento e soluções para diversos problemas em qualquer linguagens de programação.

Outra grande dificuldade foi o desenvolvimento de *PushNotifications*³ em que o usuário recebe notificações no celular mesmo que o aplicativo não esteja aberto. Passei praticamente um mês tentando desenvolver essa funcionalidade entre a alternativa nativa iOS, seguindo vários tutoriais, e depois a alternativa de *Firebase Cloud Functions*, oferecida pelo próprio servidor de banco de dados do Google. Ambos acabaram em tentativas frustradas em desenvolver essa funcionalidade tão crucial para aplicação, pois sem ela os instrutores não seriam notificados de novas dúvidas e teriam que olhar seus celulares de tempos em tempos para ver o andamento da fila, tornando-se uma alternativa inviável já que o celular é também uma ferramenta de trabalho para desenvolvedores.

Essa última dificuldade acabou impactando diretamente a adesão dos professores, além de também acabar não sendo interessante aos alunos, já que a maioria dos professores não usavam a solução. Ainda assim consegui disponibilizar o protótipo no *TestFlight*⁴[37] e consegui uma adesão de 2 professores e 20 alunos, embora não usando regularmente. Essa questão acabou impactando na minha decisão de descontinuar o projeto momentaneamente para trabalhar em outros aplicativos.

3.2. FASE 2: Desenvolvimento em equipe para AppleTV

Motivação: Em abril de 2018, foi proposto no curso particular de desenvolvimento de aplicativos que pensássemos em um aplicativo para ser desenvolvido na plataforma da *AppleTV*⁵.

³ PushNotification: Estilo de comunicação baseado na Internet onde uma transação é iniciada pelo servidor.

⁴ TestFlight: É um serviço online para instalação virtual e testes de aplicativos, pertencente a Apple Inc e oferecido pela plataforma de desenvolvimento iOS.

⁵ AppleTv: É um reproduutor de mídia digital produzido pela Apple Inc.

Apresentei a ideia do *Fila Fácil* para alguns colegas, que se interessaram em dar continuidade ao projeto.

Validação: Antes de investir tempo desenvolvendo um produto, precisávamos avaliar se existia potencial para crescer e virar uma *StartUp*⁶[2, 7]. Sendo assim buscamos criar um plano de negócios de acordo com a descrição de Steve Blank[8] para *StartUps*, ou seja, que fosse financeiramente viável, reproduzível em vários ambientes, como diferentes tipos de escolas, e por último, escalável, isto é, que pudesse impactar o máximo de pessoas sem necessitar grandes modificações de *software* ou *design*.

Para melhor estruturar o plano de negócios usamos o *BMC (Business Model Canvas)*⁷[1, 9] onde preenchemos os campos: Proposição de valor, Segmento de clientes, Relacionamento, Canais, Parcerias chave, Atividades chave, Recursos chave, Estrutura de custos e Fontes de receita. Nas primeiras versões do canvas usamos ferramentas *online* como *Canvanizer*[6] a qual ajudou a preencher cada campo corretamente. Abaixo segue uma imagem do nosso *canvas* atual, ainda sujeito a alterações, de acordo com entrevistas com novos usuários e possíveis clientes.



Business Model Canvas

1. Proposição de Valor

Precisamos estabelecer quais os principais benefícios que o produto irá proporcionar aos clientes. No nosso caso, focamos nos seguintes valores:

- Aumentar a interação e a colaboração entre a comunidade de alunos e professores;

⁶ Startup: É uma empresa emergente com o objetivo de desenvolver um modelo de negócio escalável, repetível, em condições de extrema incerteza

⁷ Business Model Canvas é uma ferramenta de gerenciamento estratégico, que permite desenvolver e esboçar modelos de negócios.

- Informar recados importantes para a turma;
- Organizar o atendimento de alunos, para que todos sejam atendidos na ordem de chegada;
- Agilizar o atendimento através de notificações para os professores de dúvidas na fila ou até mesmo com o atendimento feito pelos próprios colegas;
- Sustentabilidade ao reduzir a quantidade de papel necessário no caso de algumas escolas que usam *Post-It* para obter o mesmo resultado.

2. Segmento de Clientes

Após entrevistar muitos alunos e professores, começamos a restringir qual seria nosso público alvo, e quais teriam interesse em adquirir o produto. Sendo assim, decidimos focar primeiramente em uma turma de alunos de um curso particular e depois expandir para cursinhos preparatórios, faculdades, escolas com ensino à distância e cursos de especialização.

3. Relacionamento com os clientes

Planejamos entrar em contato com instituições que possam ter interesse no produto, mostrando dados baseados nos meses de testes realizados no curso particular ao qual disponibilizamos o aplicativo. Assim poderemos apresentar métricas, casos de sucesso e testemunhos de usuários, além de garantir que o produto já está estável e sem *bugs*⁸.

4. Canais de comunicação com os clientes

Ao oferecer o produto usaremos a própria *AppStore*⁹[10] usando modelo de negócio *White Label*¹⁰[11], onde customizamos o produto de acordo com a vontade do cliente e ele administra o acesso dos alunos na plataforma.

No caso de entrar em contato com as instituições, queremos primeiramente conversar com os professores da unidade para ver se sentem a necessidade do produto e se gostariam de experimentar gratuitamente. Assim podemos contar com a ajuda dos professores para convencer a instituição de ensino a adquirir o produto.

5. Parcerias Chave

Importante parceiro para o sucesso do projeto seria o próprio curso particular de desenvolvimento que continua ajudando na validação de novas funcionalidades e a reportar *bugs*, além de ceder o espaço e recursos para a equipe de desenvolvedores.

⁸ Falha ou defeito no programa que pode causar comportamentos inesperados, como erros ou comportamento indesejados.

⁹ App Store é uma loja virtual de programas para os equipamentos da Apple.

¹⁰ White Label é um modelo de negócio em que um produto ou serviço desenvolvido por determinada empresa pode ser revendido por outras empresas sem divulgação dos direitos autorais.

6. Atividades Chave

Apresentação para outros campi do curso particular de desenvolvimento em outras cidades, devido à semelhança com o projeto que já temos rodando com um grupo de alunos em uma das filiais desse mesmo curso particular, planejamos ainda participar e apresentar em eventos importantes na área de ensino.

7. Recursos Chave

No momento é extremamente necessário o uso de duas *AppleTV*, uma para validação da versão de produção e outra para desenvolvimento e teste antes de disponibilizar para produção. Outro recurso importante são os próprios iPhones dos desenvolvedores, onde toda base do produto é desenvolvida, chaveando entre versão de teste e produção, dependendo da necessidade.

8. Estrutura de Custos

No momento não temos nenhum custo envolvido devido à disponibilidade de recursos pelo próprio curso particular de desenvolvimento que estamos cursando. No entanto, no final do ano, toda a equipe irá concluir o curso, assim teremos que buscar recursos próprios, como *AppleTV*, tanto para desenvolvimento como para disponibilizar para o cliente, *Apple Watch* para incrementar o produto nessa plataforma, também é preciso contabilizar as horas de trabalho do grupo.

9. Fontes de Receita

O projeto ainda está em fase de teste, com a expectativa de ter uma versão comercial até o final do ano, logo não possuímos nenhuma fonte de receita. Uma fonte de receita que estamos querendo negociar é oferecer o produto para as filiais desse curso particular em troca de continuarem a ceder os recursos e o espaço para os desenvolvedores, como ocorre atualmente.

Prototipação: Ao mesmo tempo que estruturamos um modelo de negócios, também continuamos o processo de entrevistas, e logo descobrimos aspectos importantes que deveríamos abordar no projeto, como avisar instrutores e alunos sobre a dúvida de um colega. Continuamos desenvolvendo *PushNotifications*[12] pois parecia o mais natural para avisos no celular. Em paralelo, começamos o desenvolvimento da plataforma para *AppleTV*, realizando um protótipo inicial apenas com a fila de dúvidas única. Assim, a *designer* da equipe realizou *sketches*¹¹ de como ficaria a tela para poder validar com instrutores e usuários antes mesmo de gastar tempo

¹¹ Um esboço ou sketch é qualquer obra em estado inicial, que se encontre incompleta porque ainda possui muito pouca informação. É o conjunto dos objetos iniciais, mais gerais e elementares da obra a ser composta.

desenvolvendo algo que precisaria de grandes alterações no futuro. Sempre buscamos validar com menor custo e tempo possível usando a metodologia de *Design Thinking*¹²[5, 13]. Seguem abaixo os *sketches* usados na validação do fluxo das telas, tanto da *AppleTv*, quanto para o iPhone.



Imagen 3: Esmoco das telas do aplicativo no iPhone e na Apple Tv

¹² Design Thinking é o conjunto de métodos e processos para abordar problemas, relacionados a futuras aquisições de informações, análise de conhecimento e propostas de soluções

O fluxo de telas ou *Wireframe* é usado para desenhar como será a interação do usuário e prever possíveis problemas de usabilidade como precisar de um número excessivo de dados ou ações pelo usuário. A técnica de *Wireframe* também serve para indentificar qual a melhor forma de coletar dados através da localização de botões, campos de texto e imagens. Além disso, também trabalha com a forma de apresentar as informações no aplicativo, analisando hierarquia das informações através de formatos, contrastes e posicionamento na tela.

Durante o desenvolvimento tivemos a visita de professores de outra unidade que conversaram com a gente e deram mais *feedbacks* e compartilharam sobre um projeto semelhante que eles estavam planejando para AppleTv. Nessa troca de experiências obtivemos mais ideias para melhorar nosso próprio produto.

No entanto, na validação foi apontado que não era gerado nenhum valor para o usuário caso não houvesse ninguém na fila, apresentando apenas uma tela em branco. Assim, pensamos em fazer um quadro de avisos para lembrar a todos de datas e eventos importantes. Logo obtivemos nosso primeiro protótipo para *AppleTV*, como visto na imagem abaixo.

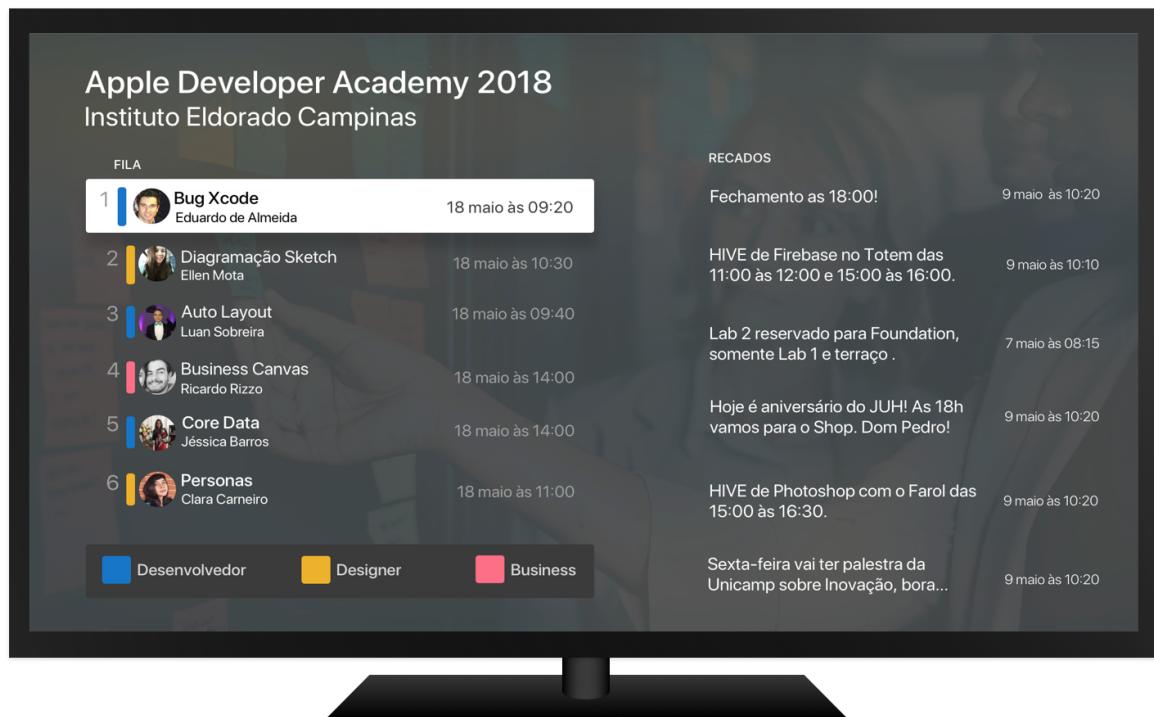


Imagen 4: segunda versão da interface do aplicativo na Apple Tv

Esse protótipo foi implementado e testado durante aproximadamente duas semanas enquanto juntávamos *feedbacks* dos alunos. Ao final dessa etapa, descobrimos que a hierarquia das informações de fila e avisos estava ruim, pois o aplicativo tem como objetivo ajudar na organização de dúvidas, mas o conteúdo principal tinha o mesmo espaço na tela que os avisos. Assim, com poucos ajustes chegamos ao *design* bem próximo do que temos hoje como produto. Abaixo segue

uma imagem com o último *design* de interface da *AppleTV*.

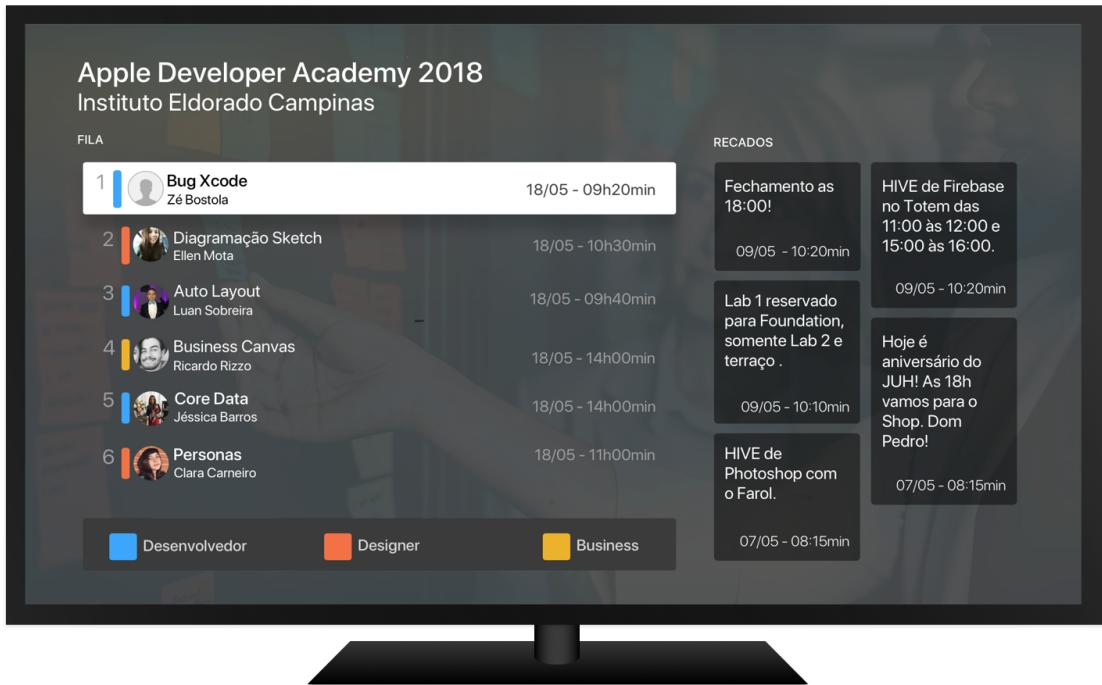


Imagen 5: Última versão do aplicativo rodando na Apple Tv.

O protótipo da versão *mobile* também sofreu alterações e *redesign* para atender melhor aos usuários, ficando com uma aparência mais profissional e moderna. Antes de realizar as alterações, foram realizados estudos para avaliar quais informações eram mais importantes. Assim foi possível reduzir a quantidade de informações da fila, dando mais ênfase para a pergunta e informações de qual aluno está com a dúvida. Abaixo estão algumas fotos das telas após o *redesign* feito durante a validação.

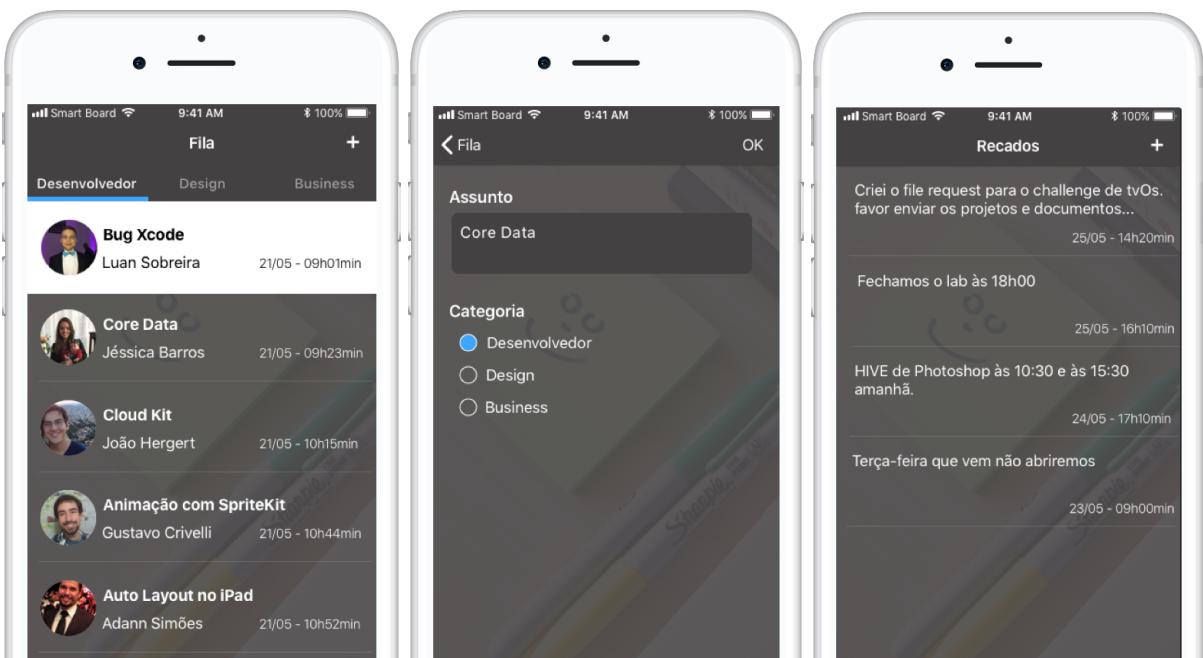


Imagen 6: Da esquerda à direita as versões finais da Tela mostrando a fila de dúvidas, tela de abertura de dúvida e Tela de avisos

Na primeira tela à esquerda é possível ver que agora as filas são divididas por área de estudo, além da fila ser composta pela foto, nome e dúvida dos alunos, com a data e hora em que foi aberta a pergunta. A foto do meio mostra a tela de abrir uma dúvida, com o campo para escrever a pergunta e depois selecionar a categoria de estudo à qual a dúvida pertence. Já a última tela mostra o quadro de avisos visto pelo celular, sendo ele o mesmo que é mostrado na *AppleTv*.

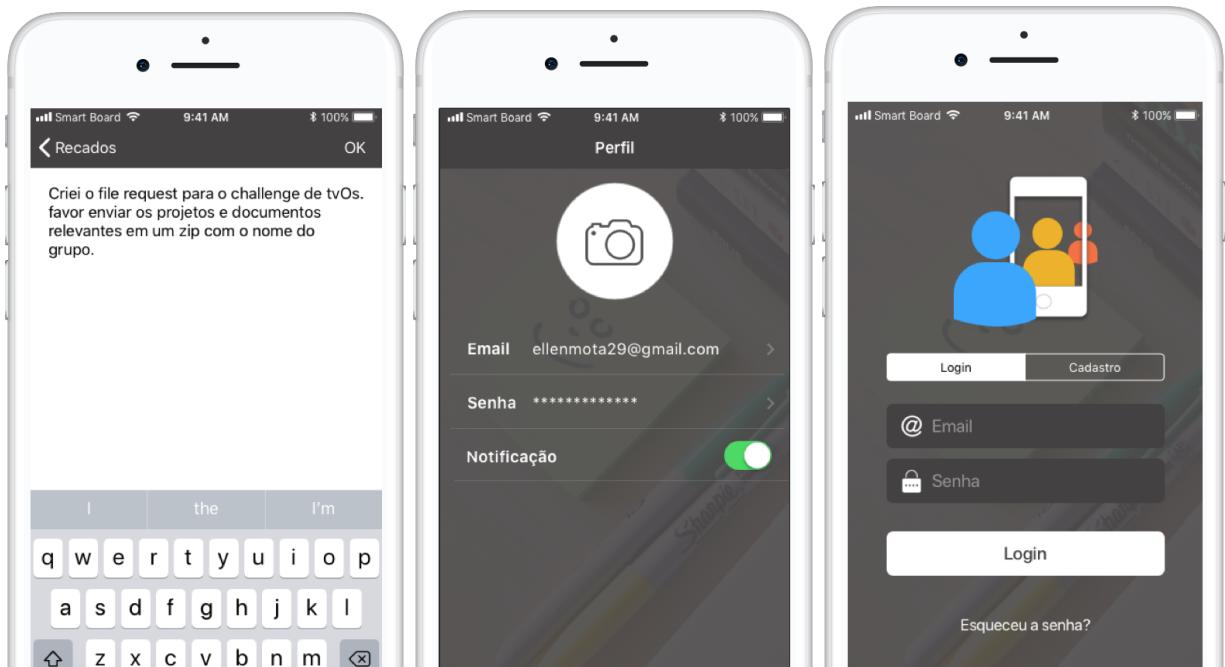


Imagen 7: Da esquerda à direita, tela de criar aviso, Tela de perfil do usuário e Tela de login.

Nas telas acima, a da esquerda mostra como se adiciona um recado para os outros alunos, a do meio mostra a tela de perfil, sendo possível ser editada pelo aluno para acrescentar uma foto, alterar senha ou e-mail, ou até mesmo desligar as notificações sobre as filas. Já a última tela mostra a tela de *login* e cadastro no aplicativo, onde novos usuários poderiam entrar no aplicativo.

Outra mudança importante foi a do logo do aplicativo, que antes contava com um ícone simples e monocromático, e passou a conter as três cores das áres de estudo do curso, além do celular como elemento principal dessa interação. Abaixo é possível observar o *design* do ícone antes e depois da atuação da *designer* do grupo. A imagem à esquerda foi a primeira versão e a imagem à direita é a versão final.

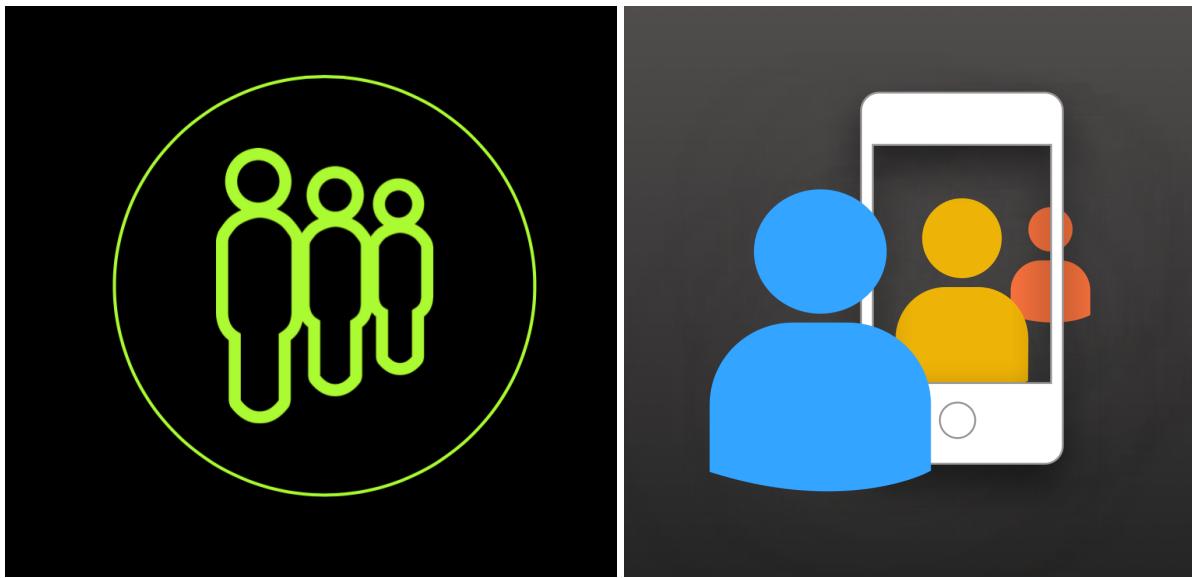


Imagen 8: Design do logotipo, a esquerda o primeiro feito por mim e a direita a versão final feita pela designer.

Dificuldades e Aprendizados: Devido à decisão de manter o banco de dados usando *Firebase*¹³[14], tivemos problemas para receber as informações do banco de dados na *AppleTv* devido à falta de suporte do *FireBase* para *TvOS*¹⁴. Assim, quase tivemos que recomeçar o banco de dados do zero para adaptar à nova plataforma e sincronizar os dados. No entanto, conversando com colegas da área, conseguimos encontrar uma solução usando o protocolo *REST.api*[15], onde a cada determinado período de tempo fazemos uma requisição ao sistema para enviar um arquivo *Json*¹⁵[16, 24] contendo toda informação do banco de dados, depois traduzimos essa informação em formato de objetos para alimentar as telas do aplicativo na *AppleTv*.

Durante o desenvolvimento em paralelo para o iPhone, tivemos muita dificuldade em implementar o *PushNotification*, o que julgamos ser vital para o funcionamento da solução. Inicialmente tentamos desenvolver usando o método nativo do sistema iOS, mas obtivemos muitos problemas e não conseguimos fazer funcionar. Logo optamos pelo serviço do próprio *Firebase* em que ele usa *Firebase Cloud Functions*[17] para programar regras em *json* para tratar os dados e saber quando e para quem ele deve enviar notificações. No entanto, descobrimos que na implementação para a *AppleTv*, não era mais tão essencial esse tipo de aviso, já que é possível estar sempre ciente da fila ao olhar a fila na TV.

Outra dificuldade que tivemos foi em lidar com o escopo do projeto, pois como tínhamos mais membros para desenvolver, idealizamos um projeto bem grande e complexo. No entanto, ao chegar mais próximo dos prazos de entrega percebemos que muitas das funcionalidades que

¹³ Firebase: é um serviço da Google de armazenamento de dados em nuvem em tempo real.

¹⁴ TvOS: Sistema operacional que funciona na Apple Tv, que faz toda interface com o usuário.

¹⁵ Json: é um formato compacto, de troca de dados simples e rápida que utiliza texto legível a humanos, no formato atributo-valor

idealizamos não teriam tempo de ser executadas, por isso tivemos que priorizar funcionalidades de acordo com o que entregaria maior valor para os usuários. A princípio tentamos seguir a metodologia ágil¹⁶[2, 18] para priorizar as funcionalidades que deveriam ser entregues, usando do Scrum¹⁷[4, 19, 25] para organizar quais tarefas deveriam ser realizadas e seus prazos.

Essa etapa também foi dificultada por problemas de comunicação no grupo inicialmente, pois existiam três canais de comunicação: *Slack*[20], *Whatsapp*[21] e *Messenger*[22], logo existiam desentendimentos ou desencontros por algum membro não ver alguma informação por não estar no canal adequado. Depois disso tivemos que estabelecer um canal oficial para realizar toda comunicação do grupo. O mesmo problema ocorreu no arquivo nas nuvens onde armazenamos todos documentos do projeto, alguns membros usavam *Dropbox*[26] para arquivos de *design*, enquanto outros usavam *Google Drive*[23] para atas de reuniões, pesquisas, *feedbacks* e outros documentos, o que gerou confusão para localizar determinados arquivos.

Um aspecto importante do desenvolvimento foram conflitos de *merge*¹⁸ entre o código fonte dos desenvolvedores, pois há arquivos que não podem ser submetidos para o versionamento¹⁹, mesmo usando arquivos no *.gitignore*²⁰ do *Github*²¹[27, 28]. Logo, precisamos adotar uma ferramenta que facilitasse a integração do código de forma limpa e organizada. Assim, adotamos o *GitKraken*²² como ferramente de visualizar e organizar o versionamento do código fonte do projeto.

3.3. FASE 3: Desenvolvimento em equipe para Apple Watch

Motivação: Novamente foi proposto um tema pelos instrutores, dessa vez era para desenvolver um aplicativo para plataforma *WatchOS*²³. Devido à incompletude do projeto em relação à plataforma *AppleTV* e a *bugs* na plataforma *iOS*, decidimos continuar o projeto para concluir e subir para testes. Para essa etapa também idealizamos o uso da *Siri*²⁴, assistente virtual do sistema *iOS*, para receber comandos como abrir uma nova dúvida, ler quais dúvidas estão em aberto ou mesmo qual sua posição na fila, para dessa forma tornar a interação com o aplicativo mais dinâmica e rápida, e ao mesmo tempo garantir acessibilidade para alunos com deficiência visual.

¹⁶ Metodologia ágil: Se consolidou nos últimos anos pela forma dinâmica, flexível e com grande aumento de produtividade.

¹⁷ Scrum: Emprega diversas ferramentas para o desenvolvimento iterativo e incremental utilizado no gerenciamento de projetos.

¹⁸ Merge: Juntar duas partes do código do mesmo projeto, unindo versões diferentes de cada desenvolvedor.

¹⁹ Versionamento: Forma de salvar em nuvem diferentes versões de um código como ponto de retorno em caso de erro.

²⁰ Gitignore: Arquivo com lista de nomes de arquivos, usado para não versionar arquivos inúteis.

²¹ Github: é uma plataforma de hospedagem de código-fonte com controle de versão usando o Git.

²² GitKraken: Plataforma de visualização de versionamento de forma gráfica.

²³ WatchOS: Sistema operacional do Apple Watch que faz a interface com o usuário.

²⁴ Siri: Assistente pessoal presente em todos os aparelhos Apple através de comandos de voz.

Validação: Ao iniciar essa etapa quisemos validar primeiramente com os instrutores a opinião deles a respeito da utilidade do *smartwatch*²⁵, pois há apenas uma parcela pequena de pessoas que possuem o aparelho no Brasil. Mantivemos, no entanto, o foco na resolução de *bugs* nas plataformas anteriores, para continuar a validação das funcionalidades que já tínhamos rodando com os alunos.

Após colher a opinião dos instrutores a respeito do que nós idealizamos para o uso do *smartwatch*, precisávamos de dados concretos para justificar o desenvolvimento e ainda precisávamos saber quais funcionalidades priorizar. Assim, iniciamos uma nova rodada de validação do *BMC* para usuários *smartwatch*. Essa etapa foi bem trabalhosa pela dificuldade de achar pessoas que se encaixassem no perfil de possível usuário. Logo, tivemos que recorrer a pesquisas na *internet*, mas ainda assim foi difícil encontrar informações sobre esse tipo de usuário.

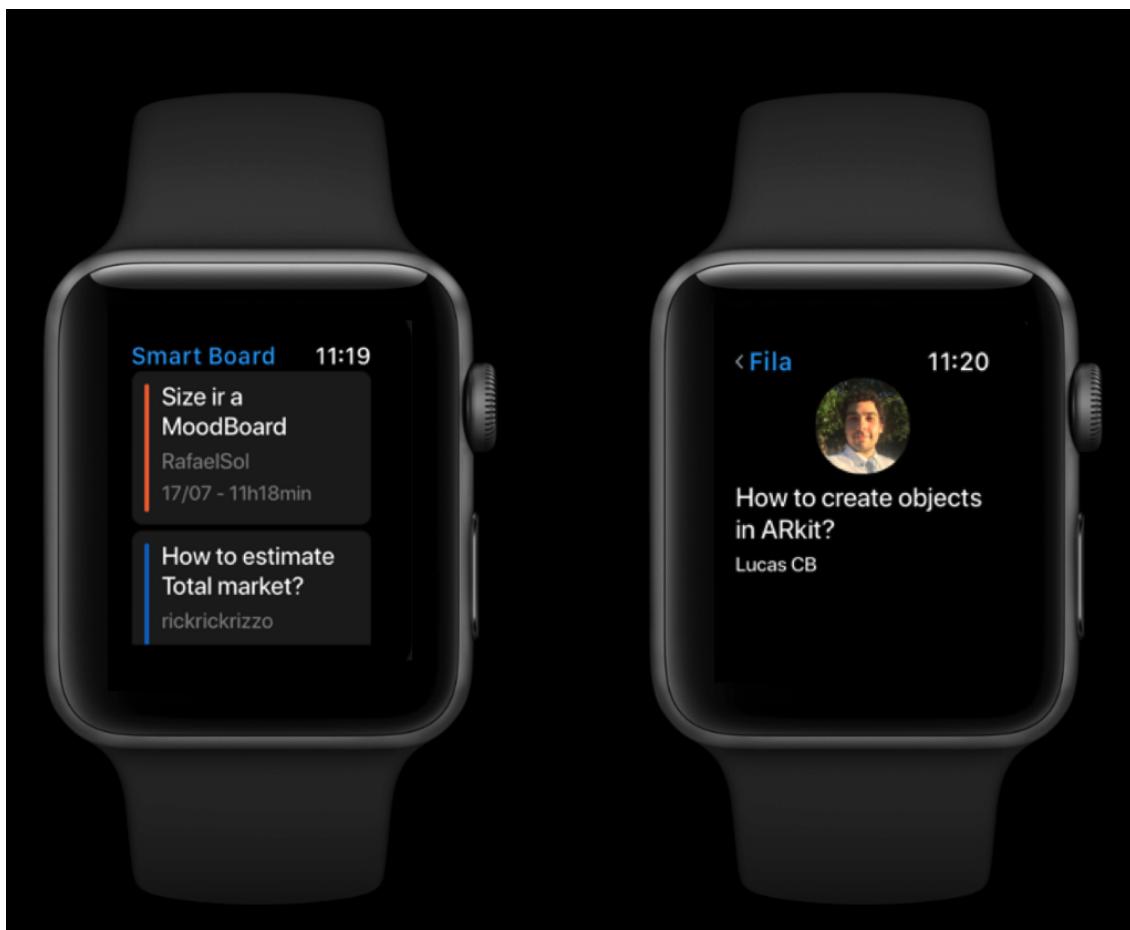


Imagen 9: Interface do aplicativo pelo Apple Watch, mostrando a fila, os detalhes da dúvida e quem perguntou.

Dificuldades e Aprendizados: Durante esse desafio, uma das desenvolvedoras do grupo

²⁵ Smartwatch: Relógio inteligente com capacidade de processamento de dados e outras funcionalidades.

resolveu sair do projeto para trabalhar com outras soluções com *AppleWatch*, o que impactou o rendimento do grupo. Depois do ocorrido, dividimos os 3 desenvolvedores restantes por frentes de trabalho. Um ficou responsável pela plataforma da *AppleTV*, outro pelo *iOS* e outro pela transferência do banco de dados para *CloudKit*[39, 40], que ainda era uma questão essencial para o sucesso do projeto. Apesar da divisão de tarefas, precisávamos nos revezar no desenvolvimento da plataforma *WatchOS*, que era a meta do mês. Isso resultou em um desenvolvimento mais lento e com menor complexidade no *WatchOS* do que nas demais plataformas.

Outra questão enfrentada foi lidar com desenvolvimento da interação com a *Siri*, que não possuia tanta documentação disponível. Assim, começamos tentando realizar comandos básicos no *iPhone* antes de tentar passar para o *smartwatch*. Ao final do mês, apesar do funcionamento quase satisfatório da *Siri*, tivemos que optar por entregar apenas o básico em *WatchOS* para focar na finalização das demais plafatormas, pois sem elas o *smartwatch* seria inútil. Durante o desenvolvimento todo, tivemos que analisar, estimar e reavaliar nossas metas para o mês, de forma que pudéssemos entregar tudo que prometemos com qualidade.

Na questão do *Firebase*, novamente tivemos problema com a integração do banco de dados devido à falta de suporte para *WatchOS*. Tivemos que adotar a mesma solução que para *AppleTv*, de usar o *REST.api*, mas com algumas resalvas pois a bateria do *smartwatch* é limitada e a transmissão de dados constantes também afeta a vida útil da carga. Assim, a atualização era menos frequente, logo precisamos usar um *CocoaPods*²⁶[30, 31] do *KingFisher*²⁷[32] para deixar imagens em *cache* e não precisar baixar com tanta frequência. Outro aspecto importante do *smartwatch* é a predominância da cor preta no *design* e o uso mínimo de imagens, pois o *smartwatch* usa ausência de luz para cor preta, para assim reduzir o consumo de energia. O tamanho da tela do *watch* também é desafiador por tentar manter a mesma linguagem das outras plataformas mas em um espaço muito menor, assim foi matida a célula da fila sem a foto, mas com a cor da área de dúvida.

Durante essa fase já tínhamos bem mais experiência com versionamento no *Github* o que reduziu muito a quantidade de erros. Começamos a aplicar algumas boas práticas em versionamento como trabalhar em *branchs* separadas para que todos pudessem trabalhar em paralelo no mesmo projeto sem conflitos. Outra ferramenta chave para auxiliar no *merge* das *branchs* foi o *GitKraken* que faz uma *interface* gráfica para o versionamento que acontece no *Github*, de forma a facilitar o entendimento da ordem das versões dos projetos.

²⁶ CocoaPods: É um gerenciador de dependencias a nível de aplicativos baseados em Objective-C e oferece um formato padrão de gerencias bibliotecas externas.

²⁷ KingFisher: É uma biblioteca disponibilizada através de CocoaPods usada para gerenciamento de cache de imagens.

Como passamos a utilizar mais o repositório para versionar o código, o projeto ganhou maior porte e assim achamos melhor manter o código privado. Sendo assim buscamos criar uma conta de estudante no *Github*, que dá direito a criar repositórios privados, no entanto, apenas enquanto se é estudante. Logo prevemos que depois de formados iremos precisar transferir o projeto para outro repositório que não cobre taxas como *GitLab* ou *BitBucket*.

Nessa etapa, devido à redução no tamanho do time, foi crucial a organização de tarefas de forma a otimizar o rendimento do grupo. Mesmo assim, grande parte do tempo foi gasto em correção de *bugs* na plataforma iOS e na finalização de funcionalidades que ainda estavam incompletas.

3.4. FASE 4: Submissão para AppStore

Motivação: Durante o mês de agosto, em contrapartida aos meses anteriores onde as plataformas eram um termo imposto, foi mantido o tema em aberto com o único requisito de publicação na loja até o fim do mês. Vimos assim a oportunidade de finalizar o projeto e começar a divulgar para outros campi desse mesmo curso particular de desenvolvimento de aplicativos.

Validação: Nessa etapa queríamos estar com o aplicativo na AppStore para deixar de usar o TestFlight, logo desde a segunda semana de desenvolvimento já começamos a realizar os requisitos mínimos para submeter á loja. Os requisitos mínimos necessários eram tirar no mínimo 3 imagens das telas, um parágrafo promocional, uma descrição do aplicativo, nome e telefone de um responsável, palavras chave para busca na loja, além de responder em quais plataformas ele estaria disponível e mais algumas perguntas sobre as tecnologias usadas no projeto. Como já tínhamos experiência de projetos passados, sabíamos que haveria dificuldades, logo planejamos submeter o aplicativo para a loja o quanto antes. Como previsto, tivemos alguns obstáculos além do trabalho de preencher as informações iniciais de forma correta para não impactar negativamente na imagem do produto para os usuários.

Antes de subir o aplicativo é necessário compilar o código de uma forma especial para mandar para loja, no entanto deve respeitar uma ordem de versionamento, ou seja não se pode subir a versão 3.0 antes da 1.0. Assim tivemos que modificar o número da versão para cada plataforma que desenvolvemos a cada nova tentativa de submissão, o que levava tempo pois a compilação necessária para a loja é lenta. Outro obstáculo que nos atrasou foi a plataforma de *Watch*. Decidimos parar o desenvolvimento momentaneamente para focar nas plataformas essenciais. No entanto, na

hora de submeter era exigido fotos de tela do *Watch* pois a plataforma ainda estava habilitada no último código compilado, logo foi preciso modificar o código novamente e atualizar o versionamento antes de poder submeter novamente, o que levou pelo menos dois dias. A última preocupação antes de finalizar a submissão foi de criar um servidor de submissão onde o avaliador da *AppStore* pudesse testar o aplicativo sem observar nossos dados de teste e nem mesmo as informações do servidor de produção.

Após submetido para avaliação, levou por volta de 4 dias para retornar com uma resposta negativa devido a alguns bugs, além de duas falhas graves. A primeira falha foi o uso do *iPhone* no logo do aplicativo já que a *Apple* veta qualquer uso da marca por terceiros, e a segunda falha foi a inconsistência no nome do aplicativo e do logotipo. A inconsistência do nome ocorreu devido à renomeação do aplicativo para *SmartBoard* na etapa anterior pela *designer* do grupo para melhor ilustrar a importância da *AppleTv* na solução, contudo após pedidos dos usuários o nome voltou a ser FilaFácil, só que alguns arquivos acabaram ficando com o nome errado. Levamos então uma semana para corrigir todos os problemas e enviar para uma nova avaliação. Nesta última submissão fomos aprovados, no entanto optamos por não deixar o aplicativo público para *download* por questão de segurança das informações e da ideia.

Após concluir essa etapa e ter um produto finalizado e pronto para demonstrações, começamos a pensar em modos de monetização²⁸, pois é necessário cobrir os custos de servidor além de obter um retorno para o esforço empregado no desenvolvimento. Logo passamos a contactar escolas e profissionais que talvez se interessassem pela solução e realizar reuniões para validar se eles enxergavam valor no produto. Assim analisamos quantitativamente e por intensidade os *feedbacks* para estimar valores, assim como pensar em modelos de monetização como assinatura mensal, com ou sem aluguel dos equipamentos necessários como *AppleTv*, ou até mesmo cobrar um taxa de acordo com o número dos alunos, mensurando pela questão da exigência sobre os servidores e na urgência em caso de manutenção.

Dificuldades e Aprendizados: Durante essa última etapa eu assim como outro membro da equipe tivemos muita dificuldade na graduação o que impactou diretamente nosso rendimento no projeto, sendo necessário reduzir horas para resolver problemas na faculdade, além de lidar com aspectos emocionais. Devido à minha leve ausência no projeto, decidi compartilhar a questão com meus colegas que foram super receptivos e se preocuparam em me apoiar na minha recuperação na graduação. Nesse momento foi realmente bom fazer parte de uma equipe unida e que teve empatia,

²⁸ Monetização: Termo derivado do verbo monetizar que designa o ato de transformar algo em dinheiro

acredito que esse seja um dos aspectos mais importantes na formação de times. Ter um bom relacionamento e gostar das pessoas com quem se trabalha aumenta as chances de sucesso. A questão das pessoas envolvidas no projeto é essencial ainda mais quando se trata da motivação da equipe, pois durante o desenvolvimento de longo prazo ocorrem momentos de altos e baixos. Como quando há muitos *bugs* para resolver e não há resultado visível, ou quando há pouco uso da solução pelos usuários, nesses momentos é importante manter as pessoas positivas, motivadas e focadas no objetivo final.

A questão do prazo de um mês fez com que otimizássemos nossa organização para conseguir realizar as tarefas dentro do prazo, apesar dos imprevistos. Tivemos também que lidar principalmente com a priorização de tarefas devido à quantidade de coisas para desenvolver em paralelo, além da manutenção do sistema e resolução de *bugs*. O pouco tempo disponível foi um dos maiores problemas pois precisamos dividir metade dos desenvolvedores refatorando o banco de dados de *Firebase* para *CloudKit* enquanto a outra metade fazia ajustes finais para submissão, além de eventuais correções de *bugs*. Em relação aos membros de *design* e *business* foi ainda mais complicado pois estes ainda estavam participando de outros grupos dividindo o tempo entre atividades de ambos grupos. Essa etapa foi realmente interessante por questão de aprender a lidar com várias tarefas ao mesmo tempo em prazo bem curto e mantendo a motivação o mais positiva possível.

4. RESULTADO

Ao finalizar o projeto pudemos observar que a ferramenta passou a ser usada diariamente e que melhorou a vida dos alunos, facilitando a iteração com os instrutores, assim como entre colegas que passaram a se ajudar mais. Ao divulgar para outras instituições fomos muito bem recebidos e convidados a implementar a solução em suas escolas. Logo começamos a trabalhar na adequação do projeto para atender novas escolas, ao mesmo tempo que continuamos a incrementar a solução já existente com novas funcionalidades para atender melhor aos nossos usuários já cadastrados.

5. TRABALHO FUTURO

Planejamos resolver os poucos *bugs* faltantes em ordem de prioridade de impacto aos usuários, à medida que recebemos novos *feedbacks*. Assim como também como já propusemos anteriormente, implementar uma forma para que os professores das instituições tenham mais controle sobre o cadastro de novos alunos, para resolver problemas de confidencialidade ou até

mesmo de conteúdo indesejado. Assim professores ou administradores poderão adicionar e remover alunos e professores, além de criar novas turmas diferentes para cada disciplina ou período.

Outra funcionalidade que prevemos é permitir ao professor criar uma organização diferente de filas de dúvidas por conteúdo, disciplinas, horários ou até mesmo por professores que possuem especializadades diferentes. Ao mesmo tempo queremos salvar as dúvidas sanadas para futuras consultas para demais alunos e até reduzir a carga do professor em caso de dúvidas recorrentes, essa funcionalidade também será importante para gerar métricas como dúvidas frequentes, quantidades de dúvidas por disciplinas ou conteúdos, ou até mesmo tempo de atendimento a cada aluno para saber qual necessita de alguma atenção especializada.

6. CONSIDERAÇÕES FINAIS

O projeto foi bem desafiador e gratificante, ao mesmo tempo que tivemos inúmeras dificuldades, também aprendemos a superar cada uma delas. Outro aspecto extremamente positivo foi ter a oportunidade de desenvolver um projeto que pode melhorar o aprendizado dos alunos.

Ao pensar no âmbito de melhorar o projeto, foi crucial estar aberto a *feedbacks* tanto positivos quanto negativos, pois o importante é fazer algo que satisfaça a necessidade do usuário e tenha impacto significativo. Assim, também foi extremamente produtiva a troca de experiências com os professores, tanto quanto com colegas alunos, para entender mais profundamente a questão a ser resolvida. A troca de ajudas técnicas também foi realmente benéfica para o andamento do projeto, pois reduzia o tempo em que nós ficávamos envolvidos em um problema.

Um dos aspectos que julgo essencial para o sucesso do projeto foi a diversidade do grupo, sendo as tarefas bem divididas entre os pontos fortes de cada membro, cada qual contribuindo de forma a complementar as habilidades um do outro. No entanto, essa diversidade ainda não teria funcionado se não fosse a organização e constante comunicação do grupo, ainda mais à medida que o projeto crescia exponencialmente. Assim, reuniões e documentação de processos, atas, *feedbacks* e pesquisas foram a base para o gerenciamento do grupo.

Ao concluir esse projeto vejo como aplicativos para celulares abrem muitas possibilidades para resolver problemas atuais de diferentes níveis de complexidade e de diversas áreas. Assim, tenho grandes expectativas com o futuro desse projeto e espero desenvolver novos projetos que beneficiem mais pessoas.

A. REFERÊNCIAS BIBLIOGRAFICAS

- [1] Alexander Osterwalder and Yves Pigneur. Business Model Generation - Inovação em Modelos de Negócios: um manual para visionários, inovadores e revolucionários. Alta Books, Rio de Janeiro, 2011.
- [2] E. Ries. The Lean Startup. Crown publishing group. Estados Unidos, 2011.
- [3] Mark H. Nichols and Karen Cator. Challenge Based Learning White Paper. Apple Inc., Cupertino, California, 2008.
- [4] K. Schwaber; J. Sutherland; D. Patel; C. Casanave; J. Miller and G. Hollowell. Scrum Development Process. Oopsia Business Object Design and Implementation Workshop, 1997.
- [5] T. Brown. Change by Design: How Design Thinking Transforms. Organizations and Inspires Innovation. Estados Unidos, 2009

B. REFERÊNCIAS WEBGRÁFICAS

- [6] <https://canvanizer.com/new/business-model-canvas> (Último acesso: 10/09/2018)
- [7] <http://ramonkayo.com/conceitos-e-metodos/o-que-e-uma-empresa-startup> (Último acesso: 20/10/2018)
- [8] <https://steveblank.com/> (Último acesso: 20/10/2018)
- [9] <https://analistamodelosdenegocios.com.br/o-que-e-o-business-model-canvas/> (Último acesso: 20/10/2018)
- [10] <https://www.telefonescelulares.com.br/o-que-e-app-store/> (Último acesso: 20/10/2018)
- [11] <https://blog.solides.com.br/o-que-e-white-label/> (Último acesso: 20/10/2018)
- [12] <https://onesignal.com/what-are-push-notifications> (Último acesso: 20/10/2018)
- [13] https://pt.wikipedia.org/wiki/Design_thinking (Último acesso: 20/10/2018)
- [14] <https://firebase.google.com> (Último acesso: 22/10/2018)
- [15] https://en.wikipedia.org/wiki/Representational_state_transfer (Último acesso: 22/10/2018)
- [16] <https://en.wikipedia.org/wiki/JSON> (Último acesso: 22/10/2018)
- [17] <https://firebase.google.com/docs/functions/> (Último acesso: 22/10/2018)
- [18] <https://br.udacity.com/blog/post/metodologia-scrum-agile> (Último acesso: 22/10/2018)

- [19] <https://www.scrum.org/resources/what-is-scrum> (Último acesso: 22/10/2018)
- [20] <https://slack.com/> (Último acesso: 22/10/2018)
- [21] <https://www.whatsapp.com/> (Último acesso: 22/10/2018)
- [22] <https://www.messenger.com/> (Último acesso: 22/10/2018)
- [23] <https://www.google.com/drive/> (Último acesso: 22/10/2018)
- [24] <https://www.json.org/> (Último acesso: 22/10/2018)
- [25] [https://pt.wikipedia.org/wiki/Scrum_\(desenvolvimento_de_software\)](https://pt.wikipedia.org/wiki/Scrum_(desenvolvimento_de_software)) (Último acesso: 22/10/2018)
- [26] <https://www.dropbox.com/> (Último acesso: 22/10/2018)
- [27] <https://pt.wikipedia.org/wiki/GitHub> (Último acesso: 22/10/2018)
- [28] <https://github.com/> (Último acesso: 22/10/2018)
- [29] <http://www.syncdev.com/minimum-viable-product/> F. Robinson. Minimum Valuable Product (Último acesso: 20/11/2018)
- [30] <https://en.wikipedia.org/wiki/CocoaPods> (Último acesso: 20/11/2018)
- [31] <https://cocoapods.org/> (Último acesso: 20/11/2018)
- [32] <https://cocoapods.org/pods/Kingfisher> (Último acesso: 20/11/2018)
- [33] <https://pt.wikipedia.org/wiki/NoSQL> (Último acesso: 20/11/2018)
- [34] <http://nosql-database.org/> (Último acesso: 20/11/2018)
- [35] https://developer.mozilla.org/en-US/docs/Web/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture (Último acesso: 20/11/2018)
- [36] <https://pt.wikipedia.org/wiki/MVC> (Último acesso: 20/11/2018)
- [37] <https://en.wikipedia.org/wiki/TestFlight> (Último acesso: 20/11/2018)
- [38] <https://stackoverflow.com/> (Último acesso: 20/11/2018)
- [39] <https://developer.apple.com/documentation/cloudkit> (Último acesso: 21/11/2018)
- [40] <https://en.wikipedia.org/wiki/CloudKit> (Último acesso: 21/11/2018)