



**ANTONIO MENEGHETTI FACULDADE - AMF
CURSO DE SISTEMAS DE INFORMAÇÃO**

DIEGO LEONARDO CHAVES RADISKE

Eventos AMF - aplicativo de informativos da instituição

RESTINGA SECA/RS

2018

DIEGO LEONARDO CHAVES RADISKE

Eventos AMF - aplicativo de informativos da instituição

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação, Faculdade Antonio Meneghetti-AMF.

Orientador: Profº. Ms. Fábio Sarturi Prass

RESTINGA SECA/RS

2018

FACULDADE ANTONIO MENEGHETTI

Diego Leonardo Chaves Radiske

EVENTOS AMF – APLICATIVO DE INFORMATIVOS DA INSTITUIÇÃO.

Trabalho de Conclusão de Curso-Monografia, apresentado como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação, Curso de Graduação em Sistemas de Informação, Faculdade Antonio Meneghetti-AMF.

⁂ Orientador: Prof. Ms. Fábio Prass



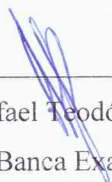
Prof. Ms. Fábio Prass

Orientador do Trabalho de Conclusão de Curso
Antonio Meneghetti Faculdade



Prof. Dr. Felipe Becker Nunes

Membro da Banca Examinadora
Antonio Meneghetti Faculdade



Prof. Dr. Rafael Teodósio Pereira

Membro da Banca Examinadora
Antonio Meneghetti Faculdade

Restinga Sêca, RS, 28 de novembro de 2018.

AGRADECIMENTOS

Após quatro anos chegou-se até este documento. Algo que só foi possível pela vontade de Deus.

Para chegar a conclusão deste projeto fora preciso muita de minha dedicação, assim, agradeço a minha persistência por algo tão desafiador em uma vida agitada.

Minha mãe por sempre estar me ajudando em todos os gestos, até nos momentos mais singelos como o café da tarde de 5 minutos, só de passada, mas que sempre tinha algo para me fortalecer.

A minha esposa que precisou ser paciente pelos diversos feriados, fins de semana, noites que passei estudando e abdiquei de sua presença em prol dos estudos.

Ao meu orientador, Ms. Fábio Sarturi Prass, que por várias disciplinas e semestres me incentivou a criar e vencer desafios, idealizador de diversos projetos que realizamos, inclusive este. Sem ele, não teria concluído este trabalho.

Ao professor Dr. Jonas Gassen, pela grande ajuda com dispositivos da Apple, emprestando a conta de desenvolvedor para meus testes e auxiliando nos momentos conflitantes.

Aos professores Dr. Rafael Pereira e Dr. Felipe Nunes, que colaboraram com sugestões e correções em meus parágrafos e no projeto.

A Majesta Inteligência Contábil que por todo o meu curso, em diversos momentos, permitiu ausentar-me da empresa em prol dos estudos.

Enfim, agradeço a todos as pessoas que fizeram parte desta etapa de minha vida seja testando o aplicativo, apresentando sugestões, falhas, ideias ou, apenas, incentivando.

“Pensar é o trabalho mais difícil que existe.
Talvez por isso tão poucos se dedicam a ele”.

Henry Ford

RESUMO

A tecnologia está presente em todos os lugares e, talvez seu uso seja maior como meio de comunicação. Assim, percebendo a necessidade de informar os integrantes da Antonio Meneghetti Faculdade (AMF) sobre assuntos relevantes da instituição e os eventos que ocorrem, podendo notificá-los através de uma ferramenta que é utilizada por todos, o *smartphone*, surge o presente trabalho tendo como objetivo o desenvolvimento de um aplicativo para dispositivos móveis, executados nos sistemas operacionais Android e iOS. No aplicativo o usuário tem acesso ao institucional, cursos, horários de ônibus, contatos, canais de comunicação e a agenda de eventos. As ferramentas que serão utilizadas para o desenvolvimento são o pacote do Android Studio, da Google e a aplicação gráfica B4X, da Anywhere Software. Como resultado, obteve-se um aplicativo para dispositivos móveis para os sistemas operacionais Android e iOS.

Palavras-chave: Android, iOS, Android Studio, B4X, B4A, B4i.

ABSTRACT

Technology is present everywhere and, perhaps, its use is greater as a means of communication. Thus, realizing the need to inform the members of Antônio Meneghetti Faculty about relevant information of the institution and the events that will occur, being able to notify them through a tool that is used by all, the smartphone, the present work appears with the objective of development of a mobile app running on Android and iOS operating systems. In the application the user will have access to the institutional, courses, bus schedules, contacts, communication channels and the agenda of events. The tools that will be used for development are the Android Studio package from Google and the B4X Graphical application from Anywhere Software. As a result, an application was obtained for mobile devices for Android and iOS operating systems.

Keywords: Android, iOS, Android Studio, B4X, B4A, B4i.

LISTA DE ILUSTRAÇÕES

Figura 1: IDE de desenvolvimento B4i.....	26
Figura 2: Visual Designer da IDE de desenvolvimento B4i/B4A.....	27
Figura 3: Configuração do compilador java a utilizar no B4A.....	28
Figura 4: Aplicativo para Android B4A-Bridge.....	29
Figura 5: Acessa a página do curso pelo aplicativo.....	32
Figura 6: Visualizar evento pelo aplicativo.....	33
Figura 7: Diagrama de classe das tabelas do aplicativo.....	34
Figura 8: Primeira versão do aplicativo.....	36
Figura 9: "Institucional" da primeira versão.....	36
Figura 10: Calendário da primeira versão.....	37
Figura 11: Versão atual do aplicativo.....	38
Figura 12: "Institucional" da versão atual.....	38
Figura 13: Cursos em sua versão atual.....	39
Figura 14: Calendário em sua versão atual.....	39
Figura 15: Tela inicial da versão final para iOS.....	46
Figura 16: Tela Cursos aberta em um iPad.....	46

Lista de tabelas

Tabela 1: Tabela comparativa entre as ferramentas de desenvolvimento.....	24
---	----

LISTA DE ABREVIATURAS

2D - Duas dimensões

3D - Três dimensões

AMF - Antonio Meneghetti Faculdade

API - *Application Programming Interface* - Interface de Programação de Aplicações

APK - *Android Package File* - Arquivo de pacote do Android

B4X - Linguagem de programação basic.

DEX - *Dalvik Executable*

IDE - *Integrated Development Environment* - Ambiente de Desenvolvimento Integrado

iOS - *iPhone operating system* - sistema operacional do iPhone

GUI - *Graphical User Interface* - Interface Gráfica do Usuário

PDF - *Portable Document Format* - Formato de Documento Portátil

SDK - *Software Development Kit* - Kit de desenvolvimento de software

SQL - *Structured Query Language* - Linguagem de Consulta Estruturada

SUMÁRIO

RESUMO.....	5
ABSTRACT.....	6
LISTA DE ABREVIATURAS.....	9
1 INTRODUÇÃO.....	12
2 OBJETIVOS.....	13
2.1 Objetivo principal.....	13
2.2 Objetivos específicos.....	13
3 JUSTIFICATIVA.....	14
4 ABORDAGEM TEÓRICA.....	15
4.1 Android.....	16
4.2 iOS.....	18
4.3 Ferramentas de desenvolvimento.....	21
4.3.1 Visão geral.....	21
4.3.2 Análise das ferramentas de desenvolvimento selecionadas.....	21
4.3.2.1 Android Studio.....	21
4.3.2.2 Eclipse.....	22
4.3.2.3 NetBeans.....	22
4.3.2.4 Xamarin.....	22
4.3.2.5 Rad Studio.....	23
4.3.2.6 B4X.....	24
4.3.2.7 Tabela comparativa.....	24
4.4 IDE de desenvolvimento B4X.....	26
5 METODOLOGIA.....	30
6 ESTUDO DE CASO.....	31
6.1 Ferramentas Utilizadas.....	31
6.2 Arquitetura da aplicação.....	32
6.2.1 Caso de uso.....	32
6.2.2 Banco de dados.....	33
6.2.3 Desenvolvimento Android com B4A.....	35

6.2.3.1 Aspectos Visuais.....	35
6.2.3.2 Códigos da estrutura.....	39
6.2.4 Desenvolvimento iOS com B4i.....	44
6.2.4.1 Aspectos Visuais.....	44
6.2.4.2 Códigos da estrutura.....	46
6.3 Limitações de contexto.....	48
6.4 Resultados.....	48
7 CONSIDERAÇÕES FINAIS.....	49
7.1 Conclusões.....	49
7.2 TRABALHOS FUTUROS.....	50
8 REFERÊNCIAS.....	50

1 INTRODUÇÃO

Os meios de comunicação vem sendo aprimorados a cada dia, tendo a disposição tecnologias cada vez mais eficientes para disseminar os dados, cobrindo quase todos os lugares. Uma instituição de ensino deve ter acesso a todos tipos de meios de comunicação disponíveis e viáveis para informar os integrantes, isso é algo fundamental.

Diante disso, tornou-se um desafio criar uma aplicação para dispositivos móveis que possibilite informar e atender as necessidades de trazer as informações da instituição a todos os envolvidos, possibilitando aos usuários terem acesso aos canais de informação e também de serem lembrados pelo aplicativo quando estes eventos ocorrerem, tornando o aplicativo um mecanismo auxiliar para que o usuário não perca os eventos que para ele forem disponibilizados.

Destarte, na educação, concordando com Saccol et al. (2010), os dispositivos móveis de comunicação estão revolucionando a educação, seja ela presencial ou a distância, que aos poucos têm visto seus métodos tradicionais serem complementados por novas metodologias mediadas pelas tecnologias móveis que definem esta aprendizagem com mobilidade como um método que permite a interação humana mesmo estando fisicamente e geograficamente distante.

Já Santiago e Mill (2009) afirmam que *“a mobilidade, traduzida em flexibilidade espaço-temporal, é pilar básico para a educação a distância (EaD)”*. Saccol et al. (2010) complementam esta afirmação salientando que as tecnologias móveis oferecem possibilidades de melhoria e redesenho dos processos educacionais em cursos virtuais, por unir estudantes e professores geograficamente separados uns dos outros e longe de espaços físicos educacionais formais.

Portanto, tento em vista que as tecnologias móveis devem estar presentes também no meio acadêmico para proporcionar a interação do

docente, aluno e instituição, através deste trabalho, pretende-se desenvolver uma aplicação que aprimore essa comunicação e fortaleça os laços de interação através das duas plataformas mais usadas, Android e iOS.

O grande desafio para o desenvolvimento do aplicativo será de estudar as tecnologias envolvidas, versões de cada sistema operacional, linguagem de programação utilizada, instabilidade da estrutura lógica de comunicação e a própria logística do sistema, seu escopo e funcionalidades. Torna-se necessário muita pesquisa técnica e específica estudando todo o contexto envolvido, procurando tornar o aplicativo o mais funcional, eficiente e prático de ser operado.

2 OBJETIVOS

Nos tópicos seguintes são abordados os objetivos principais e específicos que levaram ao desenvolvimento deste trabalho que visa desenvolver um aplicativo para dispositivos móveis para a instituição.

2.1 Objetivo principal

Implementar uma solução para a instituição no que compreende a comunicação dos valores da faculdade, informações sobre horários de ônibus, meios de comunicação, acessos a páginas da instituição e comunicado de eventos através de dispositivos móveis.

2.2 Objetivos específicos

- ✓ Proporcionar, através de uma aplicação da instituição, a informação para dispositivos móveis.

- ✓ Disponibilizar ao usuário informações sobre a instituição, seus valores, meios de contato, horários dos ônibus que os alunos utilizam e os acessos as mídias sociais.
- ✓ Alertar aos usuários, através de um calendário, os eventos que ocorrerão na instituição.
- ✓ Possibilitar aos gestores da instituição meios de informar aos usuários os eventos que ocorrerão, seus detalhes, informações dos cursos.

3 JUSTIFICATIVA

O projeto tem como objetivo desenvolver um aplicativo para dispositivos móveis que possa disponibilizar aos usuários diversas informações sobre a instituição, possuindo um calendário de eventos populado de forma dinâmica, sincronizado pela internet e com baixo tráfego de dados.

Vendo ser redundante entre alunos e professores as perguntas sobre os eventos que são marcados na instituição, é notório ser de grande valia ter uma forma de disponibilizar aos que fazem parte da instituição essas informações. Para tornar isso possível, deverá ser através de uma ferramenta que todos utilizam, os *smartphones*.

Essa conclusão pode ser observada quando levado em consideração a 29ª Pesquisa Anual de Administração e Uso de Tecnologia da Informação nas Empresas, realizada pela Fundação Getúlio Vargas de São Paulo (FGV-SP) onde informa que o Brasil superou a marca de um *smartphone* por habitante e hoje conta com 220 milhões de celulares inteligentes ativos (ESTADÃO, 2018).

Dessa forma, será desenvolvido um aplicativo para os sistemas operacionais Android e iOS que contemple as necessidades visualizadas para este projeto.

4 ABORDAGEM TEÓRICA

A Internet mudou a forma de comunicação no mundo. Está cada vez mais ao nosso redor, influenciando em nossa vida em todos os aspectos como político, econômico, lazer, investigação, comércio, serviços on-line, educação, enfim, em praticamente todas as áreas da sociedade.

Na educação, a internet abriu grandes caminhos, possibilitando buscar as informações contidas em bibliotecas, universidades, livrarias, pesquisadores de renomadas instituições dos mais variados cantos do mundo.

Diversos autores têm chamado à atenção para a importância da Internet na educação há mais de 20 anos. Começando por Gokhale (1995, apud Moura, 1998 p.129-177) que considera que a aprendizagem colaborativa dá aos alunos a oportunidade de entrar em discussão com os outros, tomar a responsabilidade pela própria aprendizagem, e assim torná-los capazes de pensamento crítico.

Ellsworth (1997, in Moura, op.cit.) observa que vive-se em uma sociedade baseada na informação, exigindo-se a capacidade de aquisição e análise dessa mesma informação. Desta forma, o mundo contemporâneo exige que o indivíduo seja capaz de pensamento crítico e capaz de solucionar problemas.

Moura, (1998) salienta que além de ser uma excelente fonte de informação, a Internet possibilita a interação com outras pessoas, ou seja, a partilha de opiniões, críticas, sugestões e visões alternativas em qualquer parte do mundo. Assim, de acordo com este autor:

“A Internet faz hoje parte do nosso mundo, incluindo o espaço escolar, e a educação não pode passar ao lado desta realidade. Este novo recurso põe à disposição um novo mar de possibilidades para novas aprendizagens, permite a interação com outras pessoas das mais variadas culturas, possibilita o intercâmbio de diferentes visões e

realidades, e auxilia a procura de respostas para os problemas. Ela é um excelente recurso para qualquer tipo de aprendizagem, em particular nas aprendizagens em que o aprendente assume o controle”.(MOURA, 1998 op.cit.).

Como desde a década de 90 tem sido visto a importância da internet no mundo acadêmico, pretende-se demonstrar a necessidade de ter uma ferramenta que proporcione uma melhor difusão da informação na AMF.

Assim, nos próximos tópicos serão apresentadas as principais características, teorias e técnicas referentes aos sistemas operacionais Android e iOS, os dispositivos móveis que os suportam e a linguagem de programação B4X (UZIEL, EREL, B4x Booklets, 2018).

4.1 Android

O Android é uma plataforma para desenvolvimento de aplicativos móveis baseada no sistema operacional Linux. Dispõe de um pacote com programas para smartphones, já com um sistema operacional, *middleware*, aplicativos e interface do usuário. É o primeiro projeto de uma plataforma *open source* para dispositivos móveis em conjunto para construir aplicações móveis inovadoras (PEREIRA; DA SILVA; 2009).

O Android causou um grande impacto quando foi anunciado, atraindo a atenção de muitos. Isso aconteceu porque por trás dele está o Google, a empresa que está revolucionando a internet. Entretanto, não é apenas o Google que está na jogada, e sim um grupo formado por empresas líderes do mercado de telefonia como a Motorola, LG, Samsung, Sony Ericsson e muitas outras. Esse grupo, chamado de Open Handset Alliance (OHA) foi criado com a intenção de padronizar uma plataforma de código aberto e livre para celulares, justamente para atender a todas as expectativas e tendências do mercado atual.

A OHA é um grupo formado por gigantes do mercado de telefonia de celulares liderados pelo Google. Entre alguns integrantes do grupo estão nomes consagrados como a HTC, LG, Motorola, Samsung, Sony Ericsson, Toshiba, HTC, Huawei, Sprint Nextel, China Mobile, T-Mobile, ASUS, Intel, Acer, Dell, Garmin e muito mais (LECHETA, 2010).

Conforme Schemberger, Freitas e Vani (2009) demonstram, o Android conta também com um SDK que disponibiliza as ferramentas e APIs necessárias para o desenvolvimento na plataforma Android, usando a linguagem JAVA.

Dentre todas as funcionalidades da plataforma, destacam-se:

- *Framework* de desenvolvimento de aplicações: reutilização de código e facilidade de acesso a recursos exclusivos e manutenção;
- Navegador web integrado: baseado no projeto *open source webkit*;
- Biblioteca de gráficos otimizada para dispositivos móveis: exclusiva biblioteca para gráficos 2D e 3D baseada na especificação OpenGL ES 1.0, com aceleração por hardware como opcional;
- SQLite: armazenamento de dados estruturados;
- Suporte multimídia: compatibilidade com os principais formatos existentes, entre eles mpeg4, h.264, mp3, .aac, .amr, .jpg, .png e .gif;

A arquitetura da plataforma Android é dividida em quatro camadas: Kernel GNU Linux, bibliotecas, *framework* para aplicações e as próprias aplicações - além da porção *runtime*, necessária para a execução dos aplicativos no dispositivo.

Segundo Schemberger, Freitas e Vani (2009), a camada mais baixa da arquitetura, Kernel Linux, é a responsável por gerenciar os processos, *threads*, arquivos e pastas, além de redes e *drivers* dos dispositivos. É responsável por gerenciar todos os processos e a memória.

Para construir aplicações para Android é utilizada a linguagem Java, independente do compilador, porém, não existe uma máquina virtual Java no

Android, o que há é uma máquina virtual chamada Dalvik, que é otimizada para execução de dispositivos móveis (LECHETA, 2010).

Ao desenvolver as aplicações para o Android será usado a linguagem Java e, quando compilado, será convertido para o formato “.dex”, que representa a aplicação do Android compilada, onde, depois disso, os arquivos “.dex” e outros recursos adicionados são compactados em um único arquivo com a extensão “.apk”, que representa a aplicação final, pronta para ser distribuída e instalada (LECHETA, 2010).

4.2 iOS

Os aplicativos para iOS são desenvolvidos usando o iOS SDK em conjunto com o ambiente de desenvolvimento Xcode da Apple. O Xcode é um ambiente de desenvolvimento integrado que codifica, compila, testa e depura os aplicativos para iOS. O ambiente Xcode também inclui um recurso chamado *Interface Builder*, que permite projetar graficamente a interface do usuário do seu aplicativo usando os componentes fornecidos pelo *UIKit Framework*. (PANOSIAN, 2017)

O Mac OS X é um moderno sistema operacional baseado em Unix desenvolvido pela Apple Inc para sua série de computadores Macintosh. O OS X é a décima versão do Mac OS. (HALVORSEN & CLARKE, 2011)

O OS X e o iOS são baseados no NeXTSTEP OS, desenvolvido pela NeXT Computer Inc, fundada por Steve Jobs depois que ele deixou a Apple em 1985. A empresa foi inicialmente financiada pelo próprio Jobs, mas depois obteve significativos investimentos externos. A NeXT foi posteriormente adquirida pela Apple e a tecnologia NeXTSTEP entrou no OS X. (HALVORSEN & CLARKE, 2011)

O iOS foi posteriormente derivado do OS X e é o sistema operacional da Apple para dispositivos móveis. Foi lançado com o primeiro iPhone, em 2007, e nesse momento foi chamado de iPhone OS, embora mais tarde foi

renomeado para iOS para refletir melhor o fato de que ele é executado em outros dispositivos móveis, como o iPod Touch, o iPad e, mais recentemente, a Apple TV. O iOS foi criado especificamente para dispositivos móveis com interfaces de toque. (HALVORSEN & CLARKE, 2011)

O núcleo do OS X e iOS, que inclui o kernel e a base Unix do sistema operacional, é conhecido como Darwin, e é um sistema operacional de código aberto publicado pela Apple. (HALVORSEN & CLARKE, 2011)

O sistema operacional Darwin executa o kernel XNU, que é baseado no código do kernel Mach, bem como em partes do sistema operacional FreeBSD. (HALVORSEN & CLARKE, 2011)

O iOS tem uma arquitetura em camadas. Entre o núcleo Darwin e o aplicativo do usuário, há um conjunto avançado de APIs de programação. O mais significativo deles é Cocoa Touch, que oferece elementos de interface gráfica especializadas para interação com o usuário baseada em toque. (HALVORSEN & CLARKE, 2011)

Outras APIs de programação incluem a API BSD, que fornece acesso a aplicativos para acesso a arquivos e dispositivos de baixo nível, bem como a API de segmentação POSIX. A camada BSD, ao contrário do Cocoa, não fornece recursos para programar aplicativos com uma interface gráfica com o usuário. O Mac OS X tem outra API importante, chamada Carbon que é uma API baseada em C que se sobrepõe ao Cocoa em termos de funcionalidade. (HALVORSEN & CLARKE, 2011)

Gráficos e multimídia são os principais diferenciais que o OS X e o iOS oferecem em relação a outros sistemas operacionais. Ambos oferecem um rico conjunto de APIs para trabalhar com gráficos e multimídia. O núcleo do sistema gráfico é o sistema Quartz que engloba o sistema de janelas, bem como o API conhecido como Quartz 2D. Quartz, baseado no modelo PDF, oferece interfaces de usuário independentes de resolução, bem como renderização de texto e gráficos sem serrilhado. A interface Extreme oferece

renderização OpenGL assistida por hardware e suportada pelo hardware gráfico.

Alguns gráficos importantes e estruturas de multimídia são listados abaixo (HALVORSEN & CLARKE, 2011):

- Quartz: Consiste na API Quartz 2D e no Quartz Compositor, que fornece o servidor gráfico de janelas. O Cocoa Drawing oferece uma interface orientada a objetos no topo do Quartz para uso em aplicativos Cocoa.
- OpenGL: API padrão do setor para desenvolvimento de aplicativos 3D. O iOS suporta uma versão do OpenGL chamada OpenGL ES, um subconjunto projetado para dispositivos embarcados.
- Core Animation: Uma API baseada em camadas integrada ao Cocoa que facilita a criação de conteúdo animado e a realização de transformações.
- Core Image: fornece suporte para trabalhar com imagens, incluindo adicionar efeitos, corte ou correção de cor.
- Core Audio: Oferece suporte para reprodução de áudio, gravação, mixagem e processamento.
- QuickTime: Uma biblioteca avançada para trabalhar com multimídia. Permite a reprodução e a gravação de áudio e vídeo, incluindo formatos profissionais.
- Core Text: uma API baseada em C para renderização e layout de texto. A API de texto de Cocoa é baseada no core text.

No seu lançamento, o iPhone OS não era capaz de executar aplicativos nativos de terceiros, mas podia rodar aplicativos da Web feitos sob medida para o iPhone, que poderiam ser adicionados à tela inicial do iPhone. Um SDK para o iPhone foi anunciado mais tarde no início de 2008, o que permitiu o desenvolvimento de aplicativos de terceiros. No entanto, a Apple exige que todos os aplicativos do iPhone sejam enviados e pré-aprovados e, portanto, assinados digitalmente, antes que um cliente possa instalá-lo por meio da App Store. (HALVORSEN & CLARKE, 2011)

4.3 Ferramentas de desenvolvimento

Nos próximos tópicos são apresentados uma visão geral e uma análise das ferramentas que foram estudadas para escolha da plataforma de desenvolvimento.

4.3.1 Visão geral

Antes de escolher a ferramenta adequada para desenvolver um aplicativo para dispositivos móveis fora necessário tomar algumas precauções. Dentre elas, para quais sistemas operacionais e até para quais tipos de dispositivos pretende-se desenvolver. Dependendo da escolha, o resultado pode ser frustrante ou fluir de forma ágil e eficiente.

Dessa forma, fora testado e estudou-se as ferramentas gratuitas Android Studio, Eclipse, NetBeans e Xamarin e as pagas Rad Studio e B4X. Uma breve descrição de cada ferramenta e os resultados estão dispostos nos itens a seguir.

4.3.2 Análise das ferramentas de desenvolvimento selecionadas

Após estudada as ferramentas de desenvolvimento, fora criado um parecer de cada ferramenta para determinar qual seria a mais adequada para realizar o projeto.

4.3.2.1 Android Studio

O Android Studio é o IDE construído pelo Google e é a opção a que tem o maior apoio da mantenedora do Android. Uma ferramenta um pouco difícil de configurar, mas está melhorando rapidamente.

No Android Studio, a codificação é feita na linguagem Java e usará referências ao Android SDK para um desenvolvimento mais básico, porém, ao necessitar opções mais avançadas, será necessário recorrer ao uso de bibliotecas e complementos onde, por vezes, pode ser necessário usar de linguagens como C++ dificultando o desenvolvimento e tornando o processo moroso (ANDROID STUDIO, 2018).

4.3.2.2 Eclipse

Nativamente usando somente o Eclipse para construir aplicativos para Android é semelhante ao Android Studio. A IDE permite criar aplicativos em Java, uso de bibliotecas e a configuração e todo o fluxo de trabalho se assemelhou muito ao Android Studio. A maior diferença está no fato que a ferramenta fora desenvolvida para outras linguagens de programação e também pode-se desenvolver para multiplataformas, tendo os compiladores corretos (ECLIPSE, 2018).

4.3.2.3 NetBeans

O NetBeans é outro IDE que suporta múltiplos idiomas e plataformas como o Eclipse. No entanto, é meio desajeitado em comparação com o Android Studio para desenvolvimento Android. Pela descrição da ferramenta, ela suporta múltiplas plataformas de desenvolvimento, entretanto, é uma ferramenta que consome muitos recursos do sistema e necessita de uma grande gama de manutenção além de exigir uma curva de aprendizagem elevada (NETBEANS, 2018).

4.3.2.4 Xamarin

O Xamarin é um IDE de propriedade da Microsoft projetado para facilitar a criação de aplicativos entre plataformas. É possível criar um

aplicativo para Android no Xamarin e depois transferi-lo para o iOS e o Windows usando a mesma base de código, IDE e APIs. Em Xamarin, a linguagem de codificação é o C# e a ferramenta possibilita a execução de testes automatizados. Há poucos meses ele tornou-se gratuita e vem junto com o pacote do Visual Studio.

Apesar de todos esses facilitadores, em testes, demonstrou-se uma aplicação pesada, de configuração complexa, consumidora de recursos do sistema e sem um fórum de discussão dedicado (XAMARIN, 2018).

4.3.2.5 Rad Studio

RAD Studio é uma IDE de desenvolvimento, mantida pela Embarcadero, que possui como linguagem o Delphi e o C++ Builder.

De acordo com a empresa desenvolvedora, o Rad Studio possibilita criar aplicativos nativos para Android e iOS, entre outras plataformas, com um único código, agilizando e facilitando o processo. Mas adverte que, na realidade, terá uma pequena quantidade de código que é exclusivo para as determinadas plataformas que ele suporta.

Apesar de ser uma plataforma paga, dispõe de uma versão demonstrativa operacional por 30 dias e atualmente fora disponibilizada a edição Community, com uma licença de uso comercial limitada (EMBARCADERO).

Em teste com a versão de demonstração, destacou-se pela sua facilidade de manipulação dos componentes e a agilidade de criar o aplicativo em testes para o sistema operacional Android. Porém, usando a versão 10.2, atual no momento do desenvolvimento, gerou incompatibilidade para compilar para processadores Intel Atom. Passado o período de avaliação sem sucesso para compilar nesse processador, as tentativas de realizar o projeto com ele fora abandonado em virtude do valor de aquisição da ferramenta.

4.3.2.6 B4X

O B4X é um conjunto de linguagens de programação BASIC para diferentes plataformas, dentre estas plataformas estão o Android e iOS, sendo o B4A para Android e o B4i para iOS. O B4A possui a versão de avaliação por 30 dias. Dessa forma, fora instalado a ferramenta e realizado testes de desenvolvimento.

Sua instalação, usabilidade com componentes e bibliotecas é simples e prático. Possui fórum muito ativo onde o próprio desenvolvedor e a equipe de desenvolvimento da ferramenta são participativos. As atualizações e as ampliações das bibliotecas são constantes.

Para desenvolver o primeiro aplicativo simples bastou apenas algumas poucas horas e executou em todos os dispositivos Android testados, inclusive com processador Intel Atom.

4.3.2.7 Tabela comparativa

Na Tabela 1 foram gerados os principais quesitos que nortearam a escolha da ferramenta mais adequada para realizar o projeto.

A tabela é resultado dos testes realizados pelo autor no momento da escolha da ferramenta de desenvolvimento e reflete, exclusivamente, a opinião do autor, seus conhecimentos com a linguagem de programação e as possibilidades financeiras disponíveis.

Tabela 1: Tabela comparativa entre as ferramentas de desenvolvimento

	Android Studio	Eclipse	NetBeans	Xamarin	Rad Studio	B4X
Linguagem	Java	Java, C/C++, php, outras	Java, C/C++, php, outras	C#	Delphi C++ Builder	BASIC
Documentação da linguagem	Sites diversos da Internet	Sites diversos da Internet	Sites diversos da Internet	Fabricante e com fórum	Fabricante com fórum	Fabricante com livros, vídeos, fórum
Documentação da ferramenta	Sites diversos da Internet	Sites diversos da Internet	Sites diversos da Internet	Fabricante e	Fabricante	Fabricante com livros, vídeos, fórum

	Android Studio	Eclipse	NetBeans	Xamarin	Rad Studio	B4X
Velocidade de desenvolvimento	Lenta	Lenta	Lenta	Média	Alta	Alta
Custo	Gratuito	Gratuito	Gratuito	Gratuito parcial	R\$ 7.240,00 a R\$ 28.958,00	B4A R\$ 357,99 B4i R\$ 357,99
Sistemas operacionais	Android	Android iOS (Com plugin)	Android iOS (Com plugin)	Android iOS	Android iOS	Android iOS
App Nativo	Sim	Depende	Depende	Sim	Sim	Sim
Complexidade de instalação	Média	Alta	Alta	Alta	Média	Baixa
Complexidade de configuração	Média	Alta	Alta	Média	Média	Baixa

Sendo a tabela uma análise do autor, cabe discutir os motivos pelos quais a escolha da ferramenta. Assim levando em conta o item Linguagem, o BASIC, C# e Delphi são as mais conhecidas, tornando essas preferíveis.

Quanto a documentação, apesar de algumas disporem de fontes em diversos locais na internet, tornam-se um ponto duvidoso como referência, justamente pela falta de um material adequado do fabricante, diferente das ferramentas que possuem a documentação oficial em sua própria página.

A velocidade de desenvolvimento é algo fundamental para escolha da ferramenta, sendo a possuidora de uma velocidade mais alta preferível.

O custo é relevante mas não definitivo. A ferramenta proporcionando uma vantagem, sendo viável, o custo torna-se um ganho em produtividade.

É indispensável que seja possível compilar, nativamente, para Android e iOS. Aquela que não permitir, já está descartada.

É, também, apreciável a facilidade de instalação e configuração. Ainda mais quando se tratar de situações que necessitem uma reinstalação ou atualização. Uma ferramenta que torna moroso esse processo, é descartada pois acarretará em possíveis falhas no aplicativo quando mal configurada.

Portanto, a partir da Tabela 1 e através da análise do autor, optou-se por utilizar a plataforma de desenvolvimento B4X, que contemplou os requisitos e seu custo acabou se tornando em ganho de produtividade.

4.4 IDE de desenvolvimento B4X

Para usar o B4A ou B4i é necessário adquirir o produto pelo site www.b4x.com, assim fora possível acesso ao fórum e bibliotecas. É uma linguagem semelhante ao Visual Basic, com suporte adicional para objetos. Aplicativos compilados pelo B4A/B4i são aplicativos Android/iOS nativos e não há *runtimes* extras ou dependências (UZIEL, 2018).

Na Figura 1 é apresentada a área de desenvolvimento do B4A e o B4i, que possuem a aparência idêntica tanto da janela de código, bibliotecas, arquivos, módulos e todas as suas características relacionadas a área destinada a criação do aplicativo.

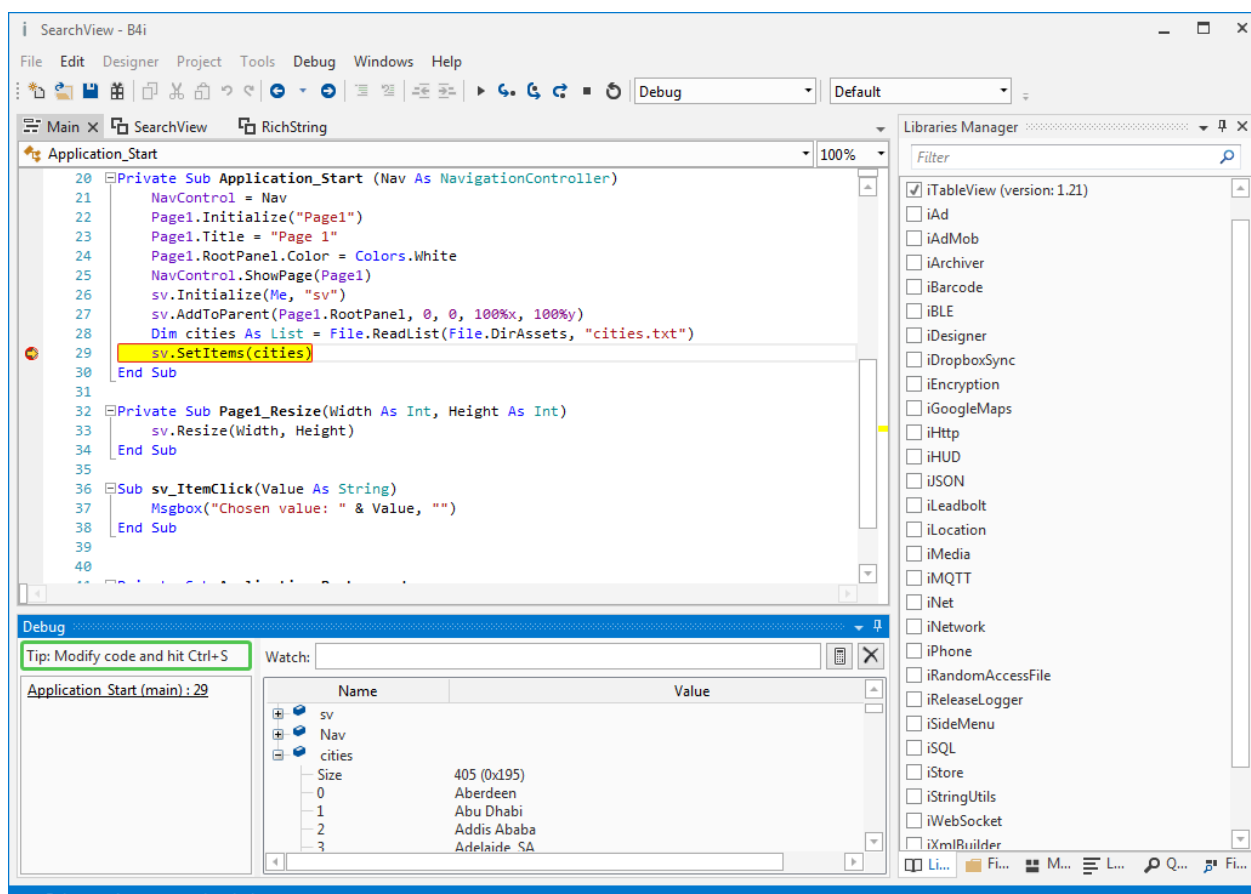


Figura 1: IDE de desenvolvimento B4i

Fonte: https://www.b4x.com/basic4android/images/SS-2015-07-08_12.24.03.png

Acessado em 18/05/2018

Na Figura 2 está sendo demonstrado o Visual Designer do B4i, idêntico ao Visual Designer do B4A e, a direita, o resultado que aparece no *smartphone*.

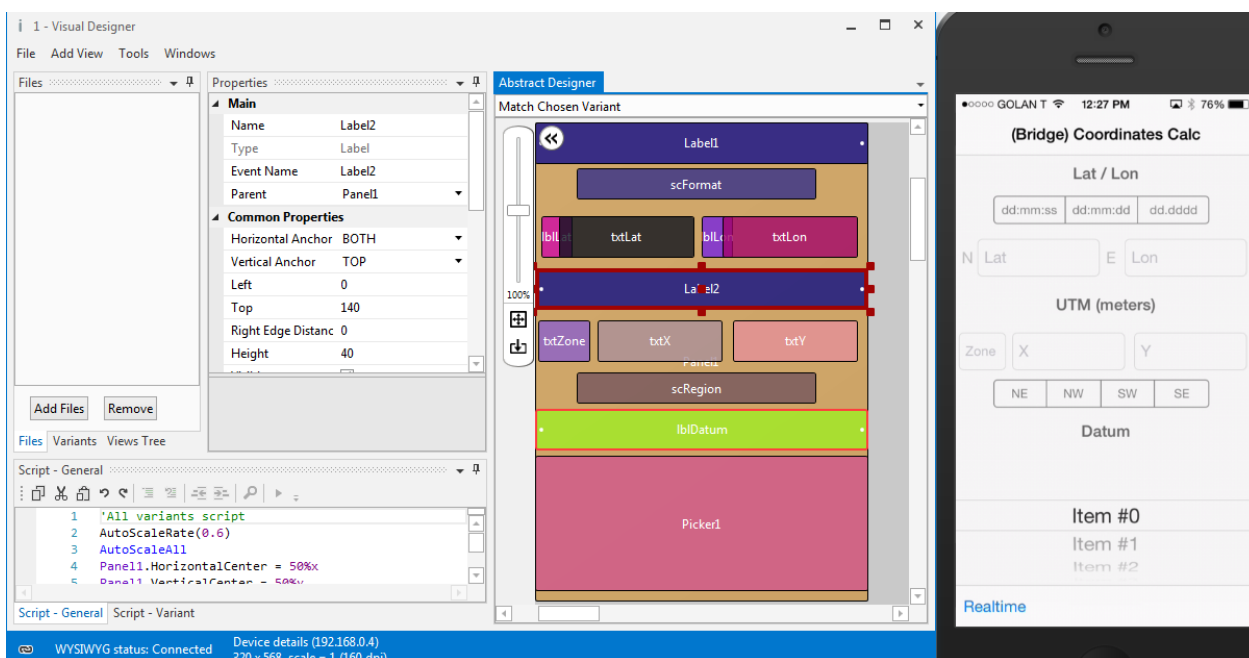


Figura 2: Visual Designer da IDE de desenvolvimento B4i/B4A

Fonte: https://www.b4x.com/basic4android/images/SS-2015-07-08_12.27.12.png

Acessado em 18/05/2018

Ao contrário de outros IDE, o B4A/B4i é 100% focado no desenvolvimento do Android/iOS, possuindo um poderoso designer GUI com suporte embutido para múltiplas telas e orientações (UZIEL, 2018).

Qualquer produto do B4X necessita de duas dependências adicionais, o Java SDK e o Android SDK, ambos gratuitos (UZIEL, 2018).

Como particularidade do Android, para utilização do Android SDK é necessário escolher a versão do API a ser instalada, Figura 3, campo `android.jar`. A quantidade de API's instalada reflete na quantidade do espaço em disco que será utilizado no sistema operacional Windows. Assim, após a instalação do Android SDK, para compilar no B4A, basta configurar para qual versão do SDK será compilado o APK (UZIEL, 2018).

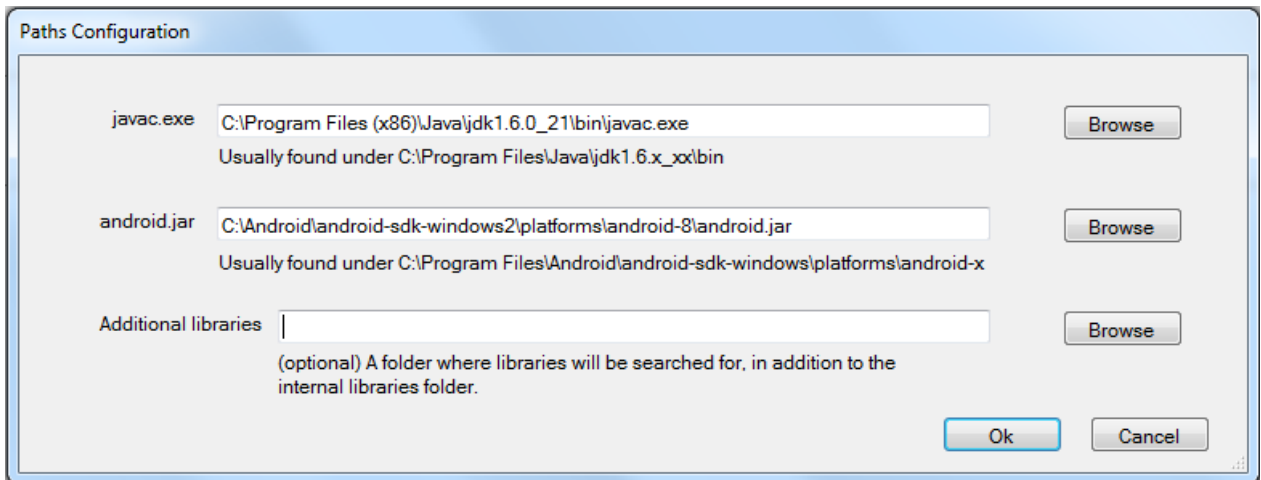


Figura 3: Configuração do compilador java a utilizar no B4A

Fonte: <https://www.b4x.com/android/forum/attachments/paths-jpg.41672/>

Acessado em 18/05/2018

Para testar a aplicação feita em B4A basta ter um dispositivo móvel com Android ou gerar o APK e instalar em um dispositivo que tenha o Android como sistema operacional.

Para o caso de ter um dispositivo móvel conectado ao sistema operacional, através de um cabo USB ou utilizando o aplicativo B4A-Bridge, é possível depurar o processo e acompanhar a interação do dispositivo com o aplicativo no dispositivo móvel, facilitando o entendimento da rotina e sendo possível analisar erros de programação ou de recursos.

Interessante ressaltar o uso do aplicativo para Android B4A-Bridge, Figura 4, que proporciona uma comunicação TCP/IP através de uma rede sem fio ou por Bluetooth com o B4A instalado no sistema operacional Windows sem a necessidade de cabos ou qualquer meio físico para instalar e testar a aplicação no Android, podendo, até, depurar o aplicativo e interagir com o código em execução (UZIEL, 2018).

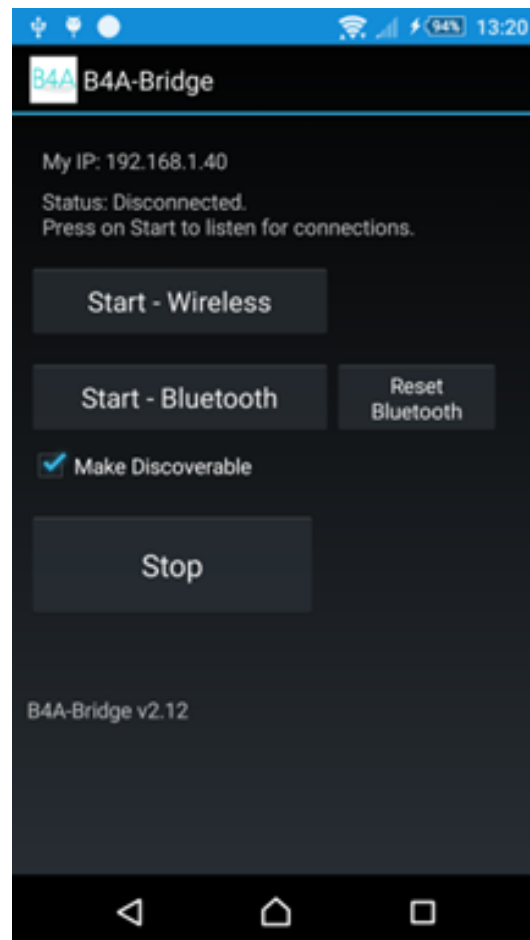


Figura 4: Aplicativo para Android B4A-Bridge

Fonte: https://www.b4x.com/android/forum/attachments/upload_2016-5-18_21-11-9-png Acessado em 18/05/2018

Diferentemente do Android, para testar a aplicação feita em B4i é necessário ter um computador com o sistema operacional OS X ou usar o *Mac Builder installation* disponibilizado pelo plano profissional do B4X além de uma conta de desenvolvedor da Apple. Mas, assim como para o B4A, também há o B4i-Bridge, que torna possível acessar e instalar o aplicativo em um dispositivo com iOS, praticamente da mesma maneira que se utiliza com Android (UZIEL, 2018).

O mesmo autor destaca ainda que dentre as diversas características do B4x, há a possibilidade de poder compartilhar o código desenvolvido para Android e iOS, facilitando, assim, o desenvolvimento multiplataforma. Como

o sistema trabalha com uso de bibliotecas, usando bibliotecas compatíveis entre as arquiteturas, pode-se compartilhar o código e compilar o mesmo projeto para um dispositivo com Android ou iOS.

Para o público brasileiro há o canal oficial <https://basic4brasil.com.br/> que proporciona a aquisição de licenças e acesso à documentação inicial. Neste há informações que são mais de 700 licenças vendidas para desenvolvedores e, de acordo com o site <http://www.communitywalk.com/B4X> há mais de 94.921 membros que utilizam o B4x em todo o mundo. De qualquer forma, para todos os usuários, no site <https://www.b4x.com> está disponível todo e qualquer acesso aos aplicativos e através do fórum o acesso às bibliotecas e documentação extra para desenvolvedores oficiais, que adquiriram a licença do sistema.

5 METODOLOGIA

De forma sucinta, o presente trabalho foi realizado em quatro etapas:

- Etapa 1. Levantamento de requisitos: Foram consultados os professores da instituição e alunos buscando o que é relevante para o aplicativo e suas funcionalidades. Utilizado o método de entrevista, buscando as necessidades e o que ensejavam que o aplicativo pudesse informar.

- Etapa 2. Fundamentação teórica: Realizado um estudo das ferramentas de desenvolvimento, conceitos, linguagens de programação, funcionalidades, custos, engenharia de desenvolvimento, praticidade e velocidade. Determinado os equipamentos e sistemas operacionais que suportarão o sistema bem como suas características físicas e lógicas.

- Etapa 3. Desenvolvimento do protótipo: Baseado nos requisitos e análises das ferramentas, fora desenvolvido as primeiras telas com funções e determinados métodos, *webservice*, banco de dados, tabelas e aparência.

Em paralelo ao desenvolvimento é realizado a engenharia de usabilidade em testes com alunos e professores.

- Etapa 4. Avaliação: O aplicativo móvel desenvolvido é avaliado em relação à utilidade, eficiência, eficácia e satisfação. É realizado um teste de usabilidade envolvendo alunos da instituição.

6 ESTUDO DE CASO

A partir desse item o trabalho fornecerá o estudo de caso selecionado com base na metodologia e ferramentas escolhidas, explicitando o desenvolvimento do aplicativo, para que no final seja exposto o objetivo final: Informar o usuário sobre a instituição e os eventos.

Para tanto, foram elencados alguns pontos como regra de negócio e requisitos:

- O acesso ao sistema será de forma gratuita.
- Não há necessidade de qualquer cadastro para utilizá-lo.
- Funcionalidades estáticas independente de ter acesso à internet.
- Para utilização do calendário com dados atualizados de forma dinâmica, será necessário acesso à internet.
- O usuário deverá acessar as funções do aplicativo em, no máximo, dois toques na tela.
- O sistema terá recursos para o usuário voltar a tela anterior.
- O aplicativo deverá ser executado em segundo plano para manter o calendário atualizado.

6.1 Ferramentas Utilizadas

Para desenvolver os casos de uso e diagrama de classe fora utilizada a ferramenta Astah Community.

No tocante ao aplicativo, o B4A e B4i do pacote B4X proporcionou todas as ferramentas necessárias para executar o desenvolvimento da interface, código da aplicação e criação do banco de dados.

O banco de dados e as tabelas do aplicativo foram gerados em uma classe, no código do B4X, sem a necessidade de ferramentas externas.

O banco de dados utilizado na aplicação mobile é o SQLite pelo fato do sistema operacional já possuir suporte nativo a ele. O SQLite é uma biblioteca desenvolvida em C e sua sintaxe padrão é SQL (SQLITE, 2014). Com ele foi possível estabelecer um padrão de dados entre o banco de dados local da aplicação mobile e o banco de dados remoto.

6.2 Arquitetura da aplicação

Neste tópico será abordado os procedimentos para desenvolvimento do aplicativo para dispositivos móveis.

6.2.1 Caso de uso

Com o Astah Community foi gerado pequenos processos que o usuário executará no aplicativo. Para qualquer opção que o usuário buscar, deve haver poucos toques até chegar ao resultado, conforme visualizado nos casos de uso da Figura 5 e Figura 6.

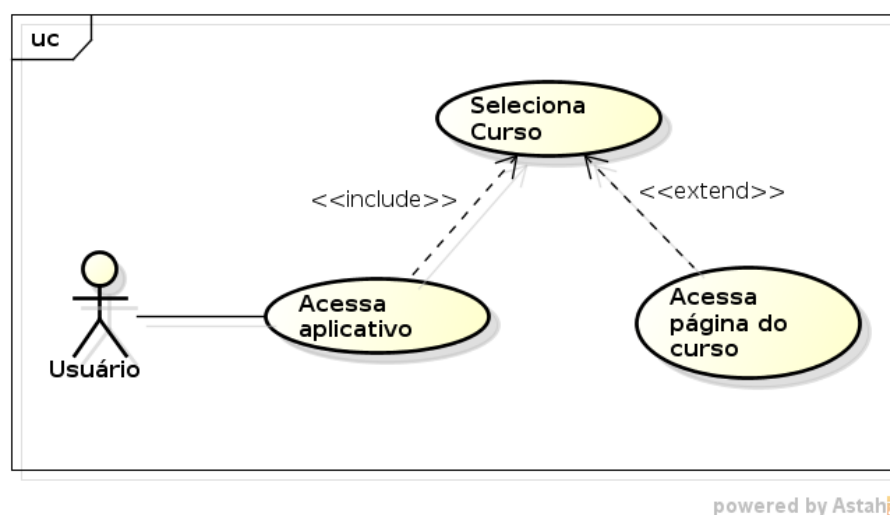


Figura 5: Acesso a página do curso pelo aplicativo

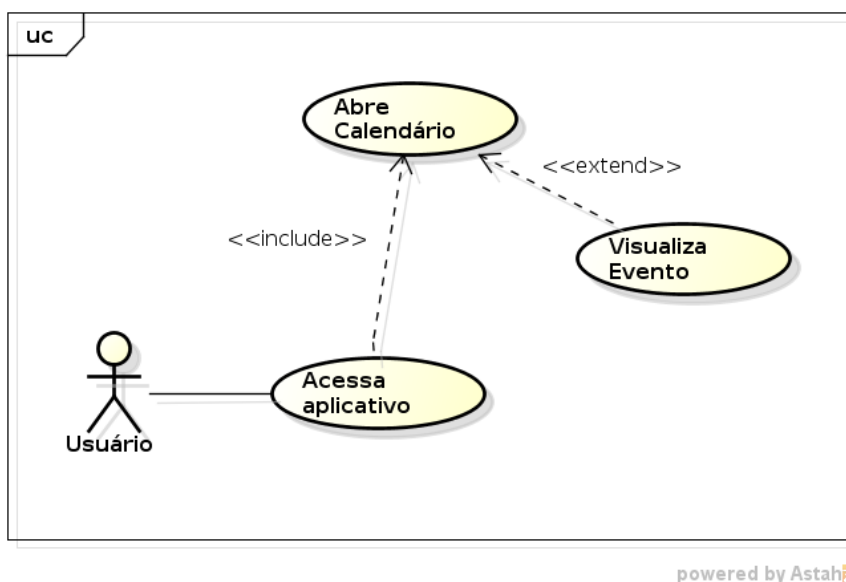


Figura 6: Visualizar evento pelo aplicativo

Dessa forma, conforme visualizado na Figura 5, quando o usuário desejar acessar a página do curso, basta acessar o aplicativo, selecionar o item curso e acionar o botão correspondente ao seu curso que será aberto o navegador na página do curso da instituição. Em 3 toques ele chegará ao local desejado.

Semelhante a visualização de eventos do dia, Figura 6, que, através do aplicativo, bastará selecionar o item Calendário e, ao escolher o dia, será carregado os eventos do dia escolhido com prévias de visualização.

6.2.2 Banco de dados

Após definir as funcionalidades previstas para o aplicativo, foi elaborado o diagrama de classe das tabelas do banco de dados.

Para tal também fora usado o Astah Community e gerado a Figura 7.

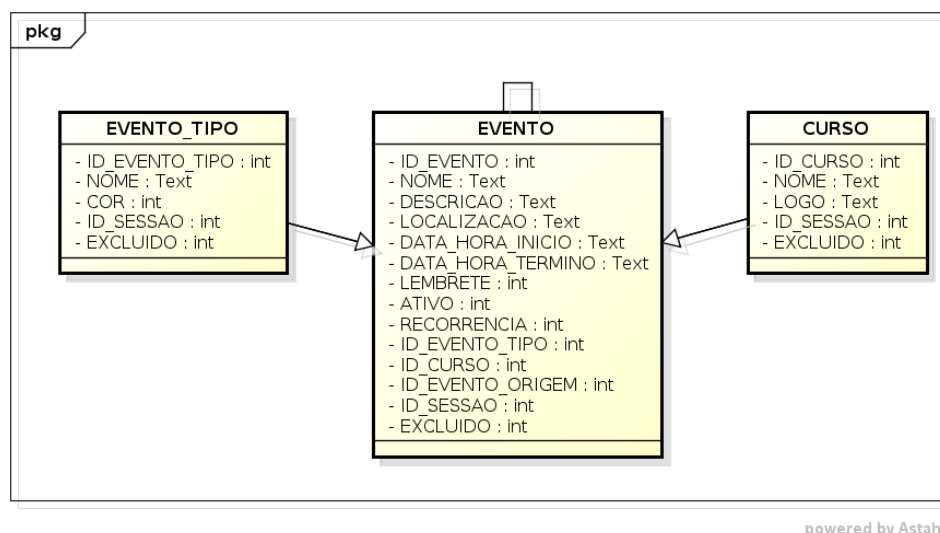


Figura 7: Diagrama de classe das tabelas do aplicativo

Com base na Figura 7, será apresentada uma breve explicação sobre as tabelas que compõem o banco de dados implementado no aplicativo.

- **EVENTO_TIPO**: responsável por armazenar os tipos de eventos que serão mostrados na tabela **EVENTO** podendo ter variação de cor e nome.
- **CURSO**: utilizada para armazenar o nome e o logotipo do curso.
- **EVENTO**: nesta tabela está armazenado todos os eventos que estarão disponíveis no calendário possuindo relacionamentos com as tabelas **EVENTO_TIPO**, **CURSO** e a própria **EVENTO**.

Como as tabelas do aplicativo são alimentadas através de uma consulta ao webservice, nelas são replicadas somente os dados necessários ao funcionamento do aplicativo móvel.

O aplicativo foi desenvolvido baseado na arquitetura cliente/servidor, onde o cliente, aplicativo móvel, estabelece uma conexão através do protocolo HTTP e então as requisições são enviadas para o servidor através do método POST. O Servidor recebe as requisições, verifica os parâmetros recebidos para saber qual base de dados está sendo referenciada e, só então, os dados são processados. Depois de obtido o resultado, os dados da

requisição são devolvidos para o módulo cliente da aplicação e então carregado no banco de dados do aplicativo cliente.

Após instalado no dispositivo móvel, em seu primeiro acesso, o aplicativo atualiza sua base de dados com o servidor. Depois da primeira carga, as próximas consultas são realizadas em tempos determinados. A verificação é realizada pela ID da última sessão e, havendo uma nova sessão, são atualizados os registros.

6.2.3 Desenvolvimento Android com B4A

Para o relato do desenvolvimento do aplicativo com o B4A, a explanação fora dividida em aspectos visuais e códigos da estrutura.

6.2.3.1 Aspectos Visuais

Baseado nas necessidades elencadas, foram desenvolvidos, separadamente, testes de como realizar os procedimentos no Android, entendendo as particularidades do sistema operacional. A cada procedimento executado com sucesso, era agregado a funcionalidade ao aplicativo.

Apesar dos testes estarem sendo executados no sistema operacional Android, ocorreu o cuidado de utilizar recursos que sejam compatíveis ou similares no sistema operacional iOS. Assim, nesse processo de aprendizagem, desenvolvimento e testes, chegou-se ao primeiro esboço do aplicativo para Android, o APK possível de ser instalado a partir da versão 19 da API, tendo o aspecto apresentado na Figura 8 e Figura 9.



Figura 8: Primeira versão do aplicativo

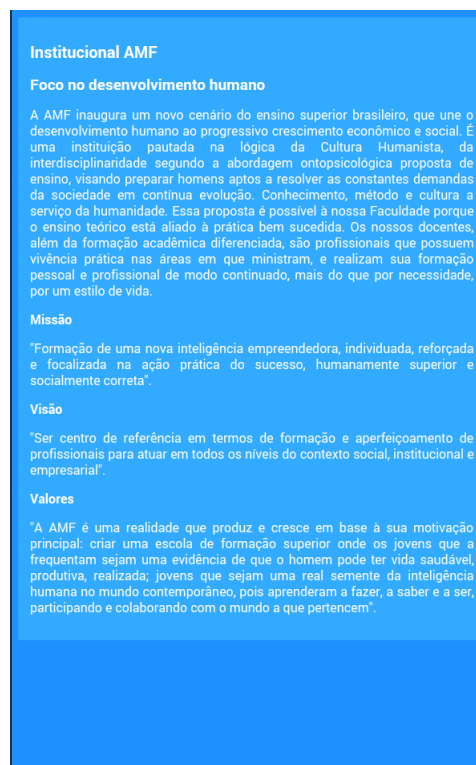


Figura 9: "Institucional" da primeira versão

Apesar de simples, já possuía cerca de 1300 linhas de código e 10 imagens. Suas funções apenas apresentavam link para página, e-mail e facebook, uma descrição da instituição, missão, visão e valores, além do calendário.

A quantidade de linhas de código inicial chegou a tanto para poder ter uma compatibilidade eficiente entre o Android e iOS no que se refere ao calendário do aplicativo, cerca de 800 linhas, desenvolvido inteiramente por código e que pode ser visto sua primeira versão na Figura 10.

Uma estrutura criada a partir de uma classe de objeto dentro do B4A, mais uma particularidade da ferramenta que gerou um aprendizado significativo para as demais funções que seriam necessárias ao aplicativo.

10 de abril de 2018						
mar	2018					mai
dom	seg	ter	qua	qui	sex	sáb
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5


10/04/2018 21:30 - 22:00
AULA DE TESTE
UMA TESTE DE EVENTO!
OPA LA IA


10/04/2018 22:00 - 22:30
VAMOS NESSA!!!
VAMOS PRA CASA!
Aqui na AMP

Figura 10: Calendário da primeira versão

No B4X é possível criar objetos visuais por código e desenvolver todo o objeto em uma classe, sem a necessidade de bibliotecas. Um processo mais trabalhoso, porém torna o objeto compatível com qualquer sistema operacional tendo em vista que a linguagem BASIC do B4X é a mesma entre as plataformas.

Deste ponto em diante, de posse das necessidades já elencadas, foram desenvolvidas as novas ferramentas do aplicativo, realizado um aprendizado sobre como melhorar o layout, sua aparência.

Aliado as indicações do orientador, estudos de outras aplicações, busca por literaturas e apreciação do usuário, desenvolveu-se novas telas, com mais recursos visuais e percebeu-se a necessidade de novas funções.

Assim, reestruturado o código em classes mais coerentes com a linguagem, definindo objetos, métodos e funções de forma adequada, chegou-se a Figura 11 e Figura 12, versão atual do aplicativo.



Figura 11: Versão atual do aplicativo

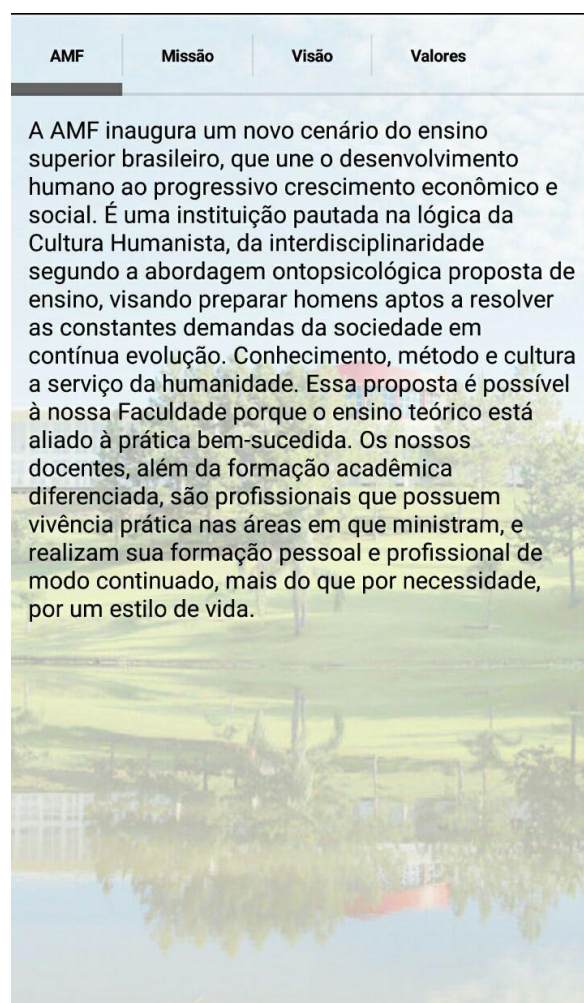


Figura 12: "Institucional" da versão atual

Demonstrando uma significativa mudança na aparência da tela principal e da Institucional, agora com os itens separadas por guias AMF, Missão, Visão e Valores, gerou-se um método para criar as novas estruturas conforme pode ser visto na Figura 13 que apresenta a tela referente ao botão Cursos e na Figura 14, com a nova visão do Calendário.

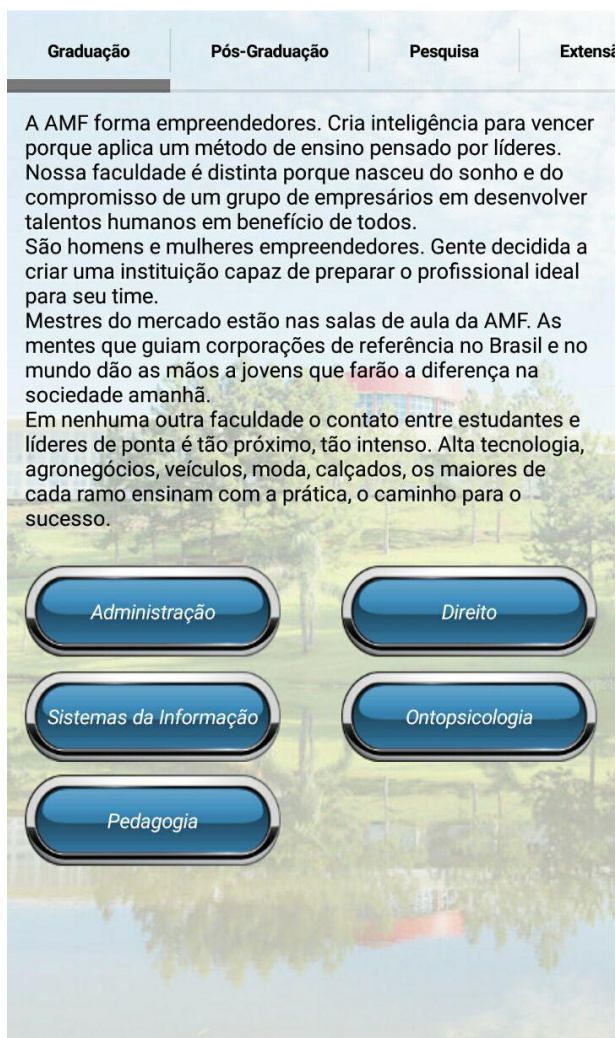


Figura 13: Cursos em sua versão atual

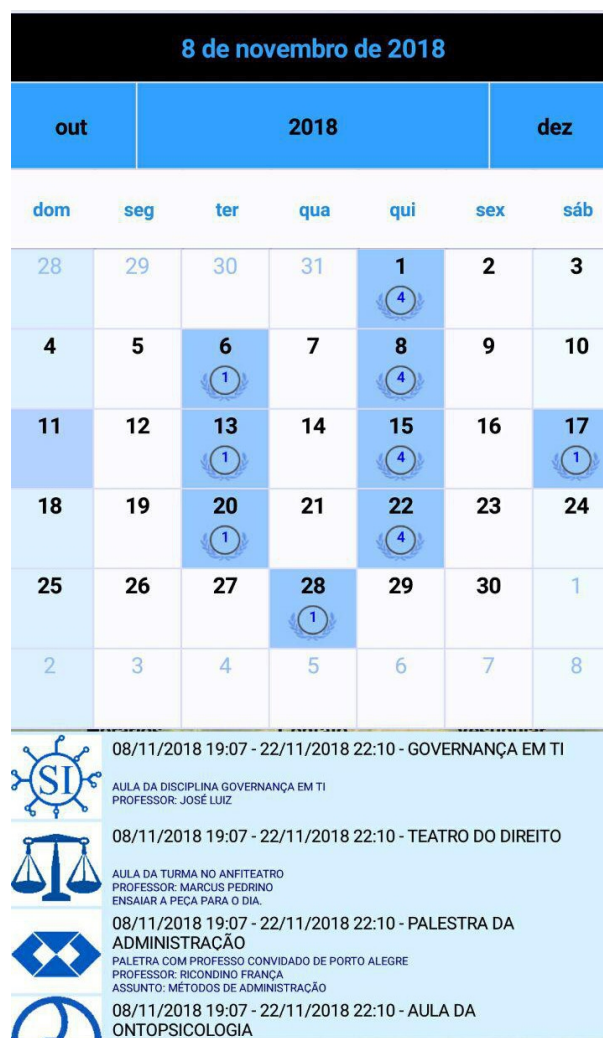


Figura 14: Calendário em sua versão atual

Criado um padrão de telas, guias, botões e textos adotados nos itens Institucional, Cursos, Horários e Contato e para o Calendário, padronizado as informações referentes a quantidade de eventos por dia, lista de eventos do dia selecionado e o logotipo de cada curso na descrição do evento que, ao selecionado na lista, trará um painel informando em detalhes o que o compõem.

6.2.3.2 Códigos da estrutura

Ao deparar-se com uma linguagem nova, um sistema operacional diferente, com dispositivos diferentes de um computador tradicional, fora

imprescindível um processo de aprendizagem, estudos na literatura e busca de respostas em fóruns. O fato de usar uma ferramenta comercial e pouco conhecida limitou a buscar ajuda somente no fórum oficial do B4X, em inglês, tendo em vista não ter encontrado outros, como acontece com a maioria das ferramentas gratuitas com as quais havia o hábito de usar.

Porém, o que aparentemente parece ser um obstáculo mostrou a qualidade que a ferramenta possui e a disponibilidade de auxiliar por parte dos usuários e, inclusive, do próprio desenvolvedor, Erel, participante ativo do fórum. Desenvolvedores de todos os continentes participam, tendo como linguagem oficial o Inglês mas também há fóruns específicos para o idioma alemão, espanhol, chinês, italiano, checo e francês. Assim, como tornou-se mais fácil a compreensão do fórum em inglês, muitas das respostas aos problemas encontrados lá foram obtidas.

Um caso emblemático enfrentado ocorreu quanto a versões do Android e suas compatibilidades referente a executar o aplicativo como um serviço em segundo plano. Até a versão 4 bastaria informar na classe de serviço a instrução

```
#StartAtBoot: True
```

na região

```
#Region Service Attributes
```

que o serviço seria adicionado nas inicializações do Android de forma permanente. Porém, a partir da versão 5 tornou-se necessário informar ao Android que o aplicativo deveria ser permanente e, a partir da versão 8, outras mudanças surgiram. Assim, para ter um serviço sendo executado em segundo plano, de forma permanente e com as permissões do usuário, um dos procedimentos foi de adicionar a instrução

```
#StartCommandReturnValue: android.app.Service.START_STICKY
```

encontrada no fórum do B4X, área do B4A, com o título “*Creating a sticky service - long running background tasks*” além de precisar estar de acordo com as mudanças dispostas nas postagens “*Starter Service -*

Consistent & Single Entry Point” e *“Automatic Foreground Mode*” sem deixar de contemplar versões anteriores já estudadas.

Portanto, entendendo esse e tantos outros detalhes sobre serviços foram criados três serviços no aplicativo sendo um de inicialização persistente, com agendamento, outro de inicialização não persistente e um serviço para quando o aplicativo estiver aberto e executado em segundo plano, até seu fechamento.

Vários detalhes de tratamento que só foram possíveis devido ao aperfeiçoamento no entendimento da plataforma e sua aceitação em trabalhar com técnicas modernas e já conhecidas como por exemplo o uso de tipos personalizados que auxiliaram na rápida execução de processos que, se fosse preciso realizar a consulta em banco de dados, reduziria a velocidade de resposta.

Como exemplo cito a criação do tipo `Proximos_Eventos` que é carregado com o serviço que inicia com o sistema e renovado em tempo determinado ou a cada atualização da tabela local. Nele está contido os próximos eventos que devem ser notificados ao usuário. Assim, a cada 50 segundos, é realizado uma verificação para saber se está no momento de emitir o lembrete baseado na lista contida em `Proximos_Eventos`.

Abaixo como é feito a declaração na sessão de processos globais.

```
Type Proximos_Eventos( _
    ID As Int, _
    Data_Hora As Long, _
    Lembrete As Int, _
    Avisado As Boolean)
```

Em outras situações adversas fora detectado divergências em tamanhos de telas nas versões do Android, estudado os padrões de telas como fazer para realizar os ajustes de textos em componentes como label (rótulo), text (caixa de texto) e button (botões).

Através de pesquisas do fórum ocorreu a criação de uma nova classe de objetos, a classe intitulada `AutoTexto` que cria uma caixa de texto, no formato de um label, com o texto ajustável ao tamanho da tela.

Com o uso de `Canvas`, uma classe do java nativo para Android, que além do desenho de primitivas, permite desenhar imagens na tela, e da biblioteca `StringUtils` do `B4A`, usada para determinar medidas e características do texto, foi criada a classe do código abaixo:

```
Sub Class_Globals
    Private cvs As Canvas
    Private mLbl As Label
    Private SU As StringUtils
    Public SizeAT As Int
End Sub

Public Sub DesignerCreateView(Base As Panel, lbl As Label, props5 As Map)
    Dim bmp As Bitmap
    bmp.InitializeMutable(1,1)
    cvs.Initialize2(bmp)
    Dim parent As Panel = Base.Parent
    parent.AddView(lbl, Base.Left, Base.Top, Base.Width, Base.Height)
    Base.RemoveView
    mLbl = lbl
    mLbl.Padding = Array As Int(0, 0, 0, 0)
    Dim jo As JavaObject = mLbl
    jo.RunMethod("setIncludeFontPadding", Array(False))
    setText(mLbl.Text)
End Sub

Public Sub setText(value As String)
    mLbl.Text = value
    Dim multipleLines As Boolean = mLbl.Text.Contains(CRLF)
    Dim Size As Float
    For Size = 2 To 80
        If CheckSize(Size, multipleLines) Then Exit
    Next
```

```

    Size = Size - 0.5
    If CheckSize(Size, multipleLines) Then Size = Size - 0.5
    mLbl.TextSize = Size
    SizeAT = Size
End Sub

Private Sub CheckSize(Size As Float, MultipleLines As Boolean) As Boolean
    mLbl.TextSize = Size
    If MultipleLines Then
        Return SU.MeasureMultilineTextHeight(mLbl, mLbl.Text) _
            > mLbl.Height
    Else
        Return cvs.MeasureStringWidth(mLbl.Text, mLbl.Typeface, Size) _
            > mLbl.Width Or _
            SU.MeasureMultilineTextHeight(mLbl, mLbl.Text) > mLbl.Height
    End If
End Sub

Public Sub getText As String
    Return mLbl.Text
End Sub

```

Esta simples classe, além de permitir criar o objeto na tela de Designer, permite criar o objeto em *runtime* e também de retornar o tamanho da fonte aplicado na caixa de texto, possibilitando o uso da medida para outros objetos.

Observando o código, torna-se notável o uso de interação direta com o interpretador Java do sistema operacional, algo permitido dentro do código do B4X em suas variantes.

Outra dentre as várias situações encontradas foi quanto ao uso de objetos existentes em um sistema operacional e inexistente em outro. Um simples objeto de lista nativo no Android, chamado de ListView no B4A não está presente no iOS, assim, buscando um objeto similar, chegou-se ao

xCustomListView. Uma classe disponibilizada no fórum que contempla a geração de uma lista e pode ser usada tanto no Android quanto no iOS.

Assim, para o Android, foram criados 3 classes de objetos, 2 classes de módulo, 1 classe de código e 3 classes de serviço, além do módulo Activity, inicial de atividade do aplicativo, totalizando 10 arquivos de codificação.

A classe de código foi utilizada para interação com o banco de dados interno. Sendo criada dessa forma, permite criar um objeto interativo referenciado, podendo ser chamado por outros processos.

Na parte de codificação do layout, criada através do Visual Designer, Figura 2, foi criado 24 arquivos de designer que, em sua maioria, contém poucos objetos básicos tendo até aqueles sem objetos. Usados separadamente devido as propriedades a eles determinadas e pelos objetos serem criados em *runtime*.

6.2.4 Desenvolvimento iOS com B4i

Após o aplicativo estar em plena operação no B4A foi o momento de adquirir a licença do B4i, licença de desenvolvedor da Apple e demais cadastros e colocar em prática o que fora aprendido. Assim para o relato do desenvolvimento do aplicativo com o B4i, a explanação fora dividida também em aspectos visuais e códigos da estrutura.

6.2.4.1 Aspectos Visuais

Com a base visual pronta, já criado no B4A, foi preciso criar o projeto em B4i e converter componentes visuais do B4A em B4i. Ambos possuem diversas bibliotecas e componentes compatíveis, mas cada qual para sua plataforma.

Assim, tratando de compatibilidade, uma biblioteca que no B4A chama-se Core, no B4i, chama-se iCore, TabStrip no BA4, iTabStrip no B4i e

assim para as demais compatíveis, bastando obtê-las no fórum do assinante. Com as bibliotecas compatíveis adicionadas o próximo passo foi tratar os arquivos de layouts feitos pelo Visual Designer.

No B4A estes arquivos têm a extensão “bal” e no B4i “bil”, possuindo uma codificação diferente, ou seja, não funciona copiar os arquivos de um para outro. Para tal, está disponível o aplicativo bal2bil, que converte os arquivos de layout do BA4 para B4i. Porém, talvez por usar bibliotecas bem recentes, não converteu algumas bibliotecas mais novas como a iTabStrip, deixando ela como um objeto Panel e, também, não converteu a AutoTexto, por ser uma classe objeto criado para o B4A e desnecessário no B4i, que tem função nativa de autoajuste de texto. Portanto, tornou-se indispensável refazer alguns layouts no Visual Designer do B4i.

O passo seguinte a compatibilidade de componentes e bibliotecas coube ao entendimento de como, de fato, um dispositivo com iOS se comporta com componentes visuais. Assim, para os testes, usou-se um iPad Air modelo A1475.

Destaca-se no sistema operacional iOS a exigência na qualidade de imagem, sendo que solicita imagens de melhor resolução. A começar pelo ícone do aplicativo que deve ser de 1024x1024 pixels.

Além desses aspectos de imagem, graças a abstração do B4X, o processo de criação dos painéis, componentes e objetos de imagem seguem idênticos entre B4A e B4i, tendo como diferença algumas propriedades entre os mesmos tipos de objetos.

Na Figura 15 está a tela principal do aplicativo aberta no iPad, utilizando as mesmas imagens em todos os objetos.

Já na Figura 16 está a mesma tela Cursos, referenciada na Figura 13, que fora aberta em um dispositivo com Android, porém, agora aberta no dispositivo com iOS.

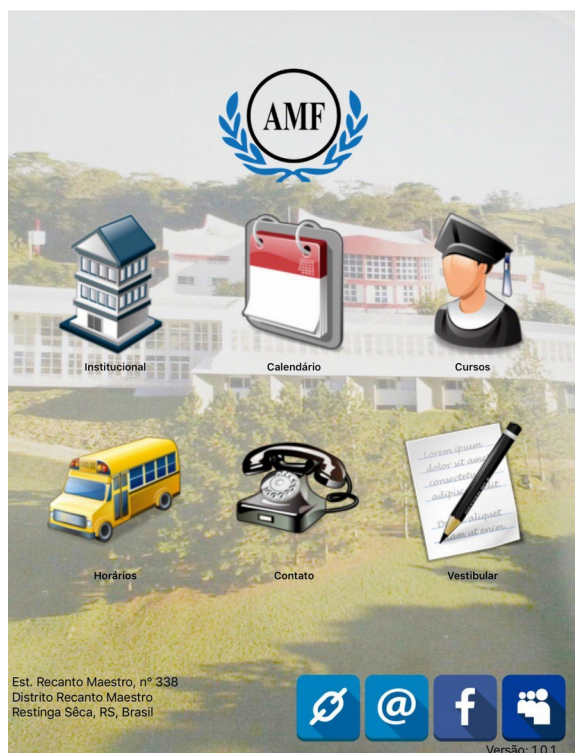


Figura 15: Tela inicial da versão final para iOS

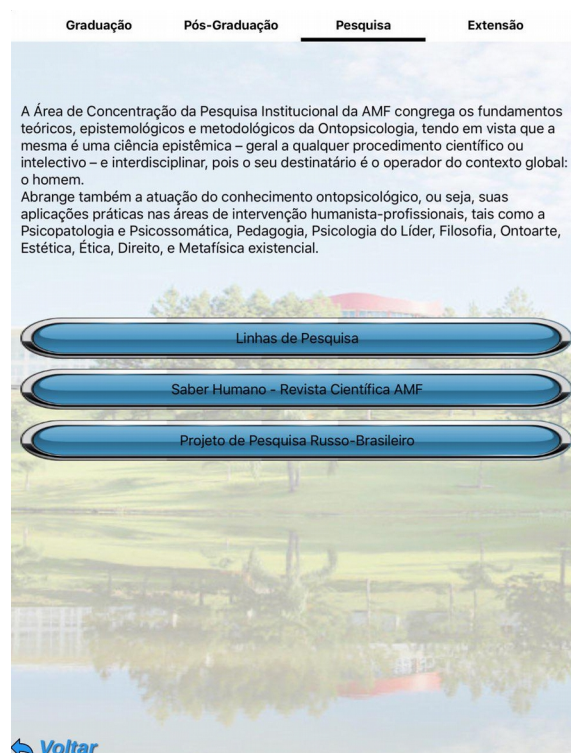


Figura 16: Tela Cursos aberta em um iPad

Um detalhe interessante a se observar nos dispositivos com iOS é que não há o botão “voltar” presente em *smartphones* com Android. Assim, adicionou-se a imagem “← Voltar” nas janelas que necessitam da funcionalidade.

6.2.4.2 Códigos da estrutura

Apesar de já estar habituado a linguagem, o mesmo não ocorre com o sistema operacional iOS que, dentre outros aspectos, possui diferenças de tratamento entre objetos, métodos e formas de execução.

Ainda que tenha feito os procedimentos para Android com métodos compatíveis, usando componentes passíveis de existência em ambos sistemas operacionais, deparou-se com propriedades divergentes entre objetos de mesma característica.

No B4A existe a biblioteca SQL, que para o B4i recebe o nome de iSQL. No momento de declarar o cursor que, em B4A declara-se

```
Dim CurAMF As Cursor
```

No B4i declara-se

```
Dim CurAMF As ResultSet
```

De acordo com o desenvolvedor, uma divergência necessária na linguagem nativa.

Mas nesse caso, o B4X possui métodos dentro do próprio código que facilitam a separação, assim, basta declarar dentro do código identificando qual procedimento pertence a qual plataforma, ficando assim:

```
#If B4A
Dim CursorAMF As Cursor
#Else If B4i
Dim CursorAMF As ResultSet
#End If
```

Procedimento muito útil no compartilhamento de código pois no caso de código compartilhado, o sistema entende a forma que deve interpretar. Apenas como exemplo, o mesmo caso ocorre quando torna-se necessário percorrer os dados da tabela. Enquanto que no B4A usa-se

```
CursorAMF.Position=Posicao
```

dentro de um loop, um for, por exemplo, no B4i não há o método *Position*, assim usa-se

```
Do While CurAMF.NextRow
```

```
... .
```

```
Loop
```

e o cursor é incrementado a cada loop.

Após acertado as divergências de código e objetos, o próximo passo foi entender como implementar processos em segundo plano no iOS já que ele não usa o conceito de serviços.

Novamente, algo que só foi possível com o auxílio do fórum. Apesar dessa divergência, ocorreu o reaproveitamento da maioria do código que

envolve os serviços no B4A, bastando refatorar e reestruturar o que envolvia os processos em ambos os projetos.

6.3 Limitações de contexto

Conforme descrito durante o desenvolvimento, há divergências entre as tecnologias de *hardware* e *software* comparando os dispositivos que podem utilizar o aplicativo, ou seja, entre *Smartphone* e *iPhone*, *iPad* e *Tablet*.

Uma das diferenças é que no Android há um componente nativo em seu hardware para realizar a função de voltar. Já o iOS é via software, necessitando adicioná-lo ao aplicativo.

Outra diferença que envolve o sistema operacional que necessitou ser tratada refere-se que o Android opera com serviços em segundo plano, já no iOS não há essa opção. Algo tratado de forma individual e que não impede o funcionamento eficaz do aplicativo, porém, caso existisse uma melhor paridade, padronizaria determinadas operações como o simples voltar de uma tela ou as formas de atuar como um serviço ou aplicativo em segundo plano.

Entretanto, o B4X e a comunidade tem abstraído diversas divergências possíveis e aquelas que, até o momento, são impossíveis de contornar, refletem, por vezes, um diferencial que torna um equipamento ou sistema desejado.

6.4 Resultados

Como resultado o projeto apresentou 1 tela principal, 20 telas secundárias, 25 links para endereços da instituição e 25 e-mails de contatos para departamentos e setores, possibilitando uma vasta gama de dados para o usuário de forma descomplicada.

O Calendário, objeto dinâmico do aplicativo, disponibiliza, além das datas, a visualização dos eventos marcados com características suficientes para orientar o usuário e avisá-lo no prazo determinado.

Assim, de acordo com os testes de usabilidade realizados com colegas em seus *smartphones*, de diferentes marcas, modelos e versões de sistema operacional, com diferentes cenários logísticos, o aplicativo comportou-se conforme havia sido proposto. Alertando o usuário, atualizando os dados, proporcionando uma visualização clara, objetiva e com poucas ações do usuário para chegar ao objetivo.

O desempenho do aplicativo, mesmo em equipamentos mais singelos, demonstrou-se rápido e sem gerar erros em todas as versões dos sistemas operacionais testados inclusive na falta de rede de comunicação sem fio, que, ao detectar sua ausência, não realiza a busca pela atualização do banco de dados.

7 CONSIDERAÇÕES FINAIS

7.1 Conclusões

Este trabalho, além de proporcionar uma ferramenta para a instituição, foi um desafio. Trabalhar com dispositivos que até o momento não eram conhecidos, sistemas operacionais que só os conhecia como usuário, plataformas e linguagens praticamente sem referências no Brasil. Algo fascinante e que, por estes motivos, proporcionou um aprendizado valioso e gerou um produto que poderá ser utilizado por todos os integrantes da instituição.

Uma ferramenta que tem condições de manter os integrantes informados de todos os eventos da instituição e, até, alertá-los com antecedência.

Ainda, a medida que o mercado de dispositivos como *smartphones* cresce, percebeu-se que as possibilidades aumentam pois somente no tempo

de desenvolvimento, novas versões de dispositivos e do sistema operacional surgiram com recursos diferenciados que possibilitariam novos meios de operação do aplicativo. É algo dinâmico e que tende a passar por modificações.

7.2 TRABALHOS FUTUROS

Para a ferramenta possibilitar uma melhor interação entre alunos da instituição, seria de grande valia a possibilidade do sistema que possui as informações do portal do aluno oferecesse um webservice para assim o aluno poder consultar notas, faltas, enfim, interagir com o portal através do aplicativo. Algo desejado até para esta versão.

Já para professores, poderia ser um portal de comunicação com o sistema de chamada e interação com o aluno. Mas para tal, seria preciso que o sistema que os professores utilizam possibilitasse uma consulta através de um webservice ou algum método semelhante.

Para ambos os casos, bastaria entrar com login e senha já cadastrados no portal que possibilitaria abrir janelas personalizadas para o usuário com seu perfil, possibilitando, ainda, carregar o calendário próprio do aluno ou professor e suas atividades como quadro de horários e avisos pessoais.

8 REFERÊNCIAS

ANDROID STUDIO; Documentação oficial do Android Studio retirada do site da empresa desenvolvedora <https://developer.android.com/studio/intro/> em 7 de outubro de 2018.

ECLIPSE; Documentação oficial do Eclipse retirada do site da Eclipse Foundation <https://www.eclipse.org/> em 7 de outubro de 2018.

EMBARCADERO; Documentação oficial do Rad Studio retirada do site da empresa desenvolvedora <https://www.embarcadero.com> em 7 de outubro de 2018.

ESTADÃO. Brasil já tem mais de um smartphone ativo por habitante, diz estudo da FGV, retirado do site <https://link.estadao.com.br/noticias/geral,brasil-ja-tem-mais-de-um-smartphone-ativo-por-habitante-diz-estudo-da-fgv,70002275238> em 11 de novembro de 2018.

HALVORSEN, Ole Henry e **CLARKE**, Douglas. OS X and iOS Kernel Programming, 2011, New York, NY, United States of America.

LECHETA, Ricardo R., Google Android, Aprenda a criar aplicações para dispositivos móveis com Android SDK. 2ª Edição, 2012, Novatec Editora Ltda. Sao Paulo, SP — Brasil.

MOURA, Rui Manuel. A Internet na Educação: Um Contributo para a Aprendizagem Autodirigida. Inovação, 11, 129-177. Disponível em <<http://members.tripod.com/RMoura/internetedu.htm> > acesso em 14 abril de 2018

NETBEANS; Documentação oficial do NetBeans retirada do site <https://netbeans.org/> em 7 de outubro de 2018.

NEUBURG, Matt, Programming iOS 11, DIVE DEEP INTO VIEWS, VIEW CONTROLLERS, AND FRAMEWORKS. Eighth Edition, 2018. O'Reilly. United States of America.

PANOSIAN, Hagop. Learn iOS Application Distribution: Successfully Distribute Apps, 2017, Hagop Panosian. New York, NY, United States of America.

PEREIRA, Lúcio; DA SILVA, Michel. Android para desenvolvedores. Rio de Janeiro, Brasil, Brasport, 2009.

SACCOL, A. Z.; et al. M-learning (mobile learning) in practice: a training experience with it professionals. Revista de Gestão da Tecnologia e Sistemas de Informação, v. 7, n. 2, 2010.

SANTIAGO, G.L.A.; MILL, D. Educação a distância e mobilidade: primeiras ações na UAB-UFSCar. In: Congresso Internacional ABED de Educação a Distância. Anais... Fortaleza: Associação Brasileira de Educação a Distância, 2009.

SCHEMBERGER, Elder; FREITAS, Ivonei; VANI, Ramiro. Plataforma Android.UNIOESTE- Universidade Estadual do Oeste do Paraná – Cascavel-PR. UNIVEL- União Educacional de Cascavel – Cascavel – PR, Brasil. 2009.

SQLITE, About SQLite; Documentação oficial do SQLite retirada do site: <https://sqlite.org/about.html> em 7 de outubro de 2018.

UZIEL, Erel; Documentação oficial do B4X retirada do site: <https://www.b4x.com/android/documentation.html> em 12 de abril de 2018.

XAMARIN; documentação oficial do xamarin retirada do site da microsoft <https://visualstudio.microsoft.com/xamarin/> em 7 de outubro de 2018.