

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DO CURSO DE GRADUAÇÃO EM TECNOLOGIA EM
SISTEMAS PARA INTERNET

LUCAS VINICIUS RUCHEL

**Aplicação móvel para automação de uma agenda: Uma forma rápida e
prática para acompanhar eventos**

TRABALHO DE CONCLUSÃO DE CURSO

TOLEDO

2017

LUCAS VINICIUS RUCHEL

Aplicação móvel para automação de uma agenda: Uma forma rápida e prática para acompanhar eventos

Trabalho de conclusão de curso de graduação, apresentada a Coordenação do Curso de Graduação em Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Roberto Milton Scheffel

TOLEDO

2017

Lucas Vinicius Ruchel

APLICAÇÃO MÓVEL PARA AUTOMAÇÃO DE UMA AGENDA: UMA FORMA RÁPIDA E PRÁTICA PARA ACOMPANHAR EVENTOS

Este Trabalho de Conclusão de Curso de Graduação foi julgado e aprovado como requisito parcial para a obtenção do Título de Tecnólogo em Sistemas para Internet, do curso de Tecnologia em Sistemas para a Internet da Universidade Tecnológica Federal do Paraná (UTFPR).

Toledo, 24 de novembro de 2017.

Prof. Fábio Engel de Camargo, Me.
Coordenador do Curso de
Tecnologia em Sistemas para a Internet

Prof. Wilson Dalle Molle, Dr.
Responsável pelos Trabalhos de Conclusão de Curso

ORIENTAÇÃO

Roberto Milton Scheffel, Me.
Universidade Tecnológica Federal do Paraná
Orientador

BANCA EXAMINADORA

Rosane Fátima Passarini, Dra.
Universidade Tecnológica Federal do Paraná

Eduardo Pezutti Beletato dos Santos, Me.
Universidade Tecnológica Federal do Paraná

AGRADECIMENTOS

A minha família por me dar suporte durante o período do curso, ajudando a superar diversas dificuldades e me dando apoio para continuar os estudos.

Ao meu orientador Roberto Milton Scheffel, pelo suporte e paciência nas correções durante o desenvolvimento deste trabalho.

A Fabiana Delabona por todo apoio durante a realização deste trabalho.

Aos meus amigos que sempre me incentivaram a estudar.

RESUMO

RUCHEL, Lucas. Aplicação móvel para automação de uma agenda: Uma forma rápida e prática para acompanhar eventos . 57 f. Trabalho de conclusão de curso – Coordenação do Curso de Graduação em Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Toledo, 2017.

Este trabalho apresenta uma análise sobre a integração de aplicativos de gerenciamento de eventos e a utilização de agendas digitais em dispositivos móveis visando integrar os horários das sessões de um evento que um usuário deseja participar com sua agenda digital. A partir da análise de aplicativos disponíveis no mercado, constatou-se que os mesmos não possuem ferramentas que permitam a integração com a agenda do usuário. Participar em múltiplos eventos requer que sejam instalados diversos aplicativos para acompanhar as diferentes programações, o que é pouco prático e pode tornar-se um processo incômodo. Devido a estes pontos observados, foi desenvolvido um aplicativo para a plataforma *Android* que, utilizando uma aplicação de gerenciamento de eventos de código-fonte aberto, sincroniza os horários dos eventos que o usuário deseja seguir com sua agenda digital, verificando se os horários estão livres ou se há conflito com outras atividades previamente agendadas. O aplicativo desenvolvido propõe uma forma de integrar os eventos que o usuário deseja participar com sua agenda digital, facilitando a visualização dos eventos que o mesmo irá participar. Em trabalhos futuros estuda-se a possibilidade de estender a aplicação, de forma a torná-la independente de um gerenciador de eventos específico, tornando-a ainda mais flexível.

Palavras-chave: Gerenciamento de Eventos, Android, Google Calendar, JAVA, Aplicação

ABSTRACT

RUCHEL, Lucas. Mobile application for calendar automation: A quick and practice way to follow events. 57 f. Trabalho de conclusão de curso – Coordenação do Curso de Graduação em Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Toledo, 2017.

This work presents an analysis of the integration of event management applications and the use of digital calendars in mobile devices, aiming to integrate the sessions of an event that a user wishes to participate with his digital agenda. From the analysis of applications available in the market, it was verified that they do not have tools that allow integration with the user's agenda. Participating in multiple events requires that several applications have to be installed to follow the different schedules, which is impractical and can become an uncomfortable process. Due to these observed points, an application for the Android platform was developed that, using an open source event management application, synchronizes the schedules of the events that the user wants to follow with his digital agenda, verifying if the schedule periods are free or if there are conflicts with other previously scheduled activities. The application developed proposes a way to integrate the events that the user wishes to participate with his digital agenda, facilitating the visualization of the events that the same will participate. In future works the possibility of extending the application is studied, in order to make it independent of a specific event manager, making it flexible.

Keywords: Event management, Android, Google Calendar, JAVA, Application

LISTA DE FIGURAS

FIGURA 1	– Modelo de funcionamento.	11
FIGURA 2	– Página inicial da aplicação <i>Open Event Organizer Server</i> , segundo modelo padrão	15
FIGURA 3	– Página com documentação da <i>API</i> utilizada	16
FIGURA 4	– Diagrama de Classes do Aplicativo Desenvolvido.	20
FIGURA 5	– Diagrama de Entidade Relacionamento do Banco de Dados do aplicativo desenvolvido.	21
FIGURA 6	– Diagrama de Casos de Uso da Aplicação de Gerenciamento de Eventos	22
FIGURA 7	– Diagrama de Casos de Uso do Aplicativo Móvel	22
FIGURA 8	– Representação de Tela de Login.	35
FIGURA 9	– Tela de Visualização de eventos.	35
FIGURA 10	– Tela de Gerenciamento de Sessões.	35
FIGURA 11	– Diagrama de Sequência para Gerenciar Sessões.	42
FIGURA 12	– Diagrama de Sequência para agendar verificação de eventos.	42
FIGURA 13	– Diagrama de Sequência para atualizar agenda do Google no dispositivo - marcar.	43
FIGURA 14	– Diagrama de Sequência para atualizar agenda do Google no dispositivo - desmarcar.	43
FIGURA 15	– Diagrama de Sequência para processar eventos sincronizados.	44
FIGURA 16	– Diagrama de Sequência para processar sessões sincronizadas.	45
FIGURA 17	– Diagrama de Sequência para sincronizar eventos.	46
FIGURA 18	– Diagrama de Sequência para sincronizar sessões.	47
FIGURA 19	– Diagrama de Sequência para verificar eventos em segundo plano.	47
FIGURA 20	– Diagrama de Sequência para verificar novos eventos.	48
FIGURA 21	– Diagrama de Sequência para informar credenciais de acesso.	49
FIGURA 22	– Diagrama de Sequência para verificar presença de <i>token</i>	49
FIGURA 23	– Tela de <i>Login</i>	51
FIGURA 24	– Tela de Visualização de Eventos Seguidos.	51
FIGURA 25	– Sessões de Evento Sincronizado.	52
FIGURA 26	– Detalhes de Sessão Adicionada.	52
FIGURA 27	– Estado sessão que usuário pretende participar.	52
FIGURA 28	– Estado de sessão que usuário não pretende participar.	52
FIGURA 29	– Notificação de Alteração de Sessão.	53
FIGURA 30	– Estado de sessão após alteração.	53
FIGURA 31	– Tela informando conflitos de horários ao inserir em agenda sessão sobrepondo outra.	54
FIGURA 32	– Tela de exibição de sessão inserida na agenda do usuário.	54

LISTA DE TABELAS

TABELA 1	–	Análise de Requisitos	19
TABELA 2	–	Fluxo de Ações para Criar Evento.	24
TABELA 3	–	Fluxo de Ações para Editar Evento.	25
TABELA 4	–	Fluxo de Ações para Remover Evento.	26
TABELA 5	–	Fluxo de Ações para gerenciar propostas de sessões para evento.	27
TABELA 6	–	Fluxo de Ações para Gerenciar oradores de sessões.	29
TABELA 7	–	Fluxo de Ações para Gerenciar Programação de Evento.	30
TABELA 8	–	Fluxo de Ações para se Cadastrar no Sistema.	31
TABELA 9	–	Fluxo de Ações para Validar Credenciais de Acesso.	31
TABELA 10	–	Fluxo de Ações para registrar interesse em evento.	32
TABELA 11	–	Fluxo de Ações para Submeter Proposta de Sessão.	33
TABELA 12	–	Fluxo de Ações para Acessar Informações do Sistema.	34
TABELA 13	–	Fluxo de Ações para Informar Credenciais de Acesso no Aplicativo Móvel.	37
TABELA 14	–	Fluxo de Ações para Verificar Novos Eventos.	38
TABELA 15	–	Fluxo de Ações para Gerenciar Sessões.	39
TABELA 16	–	Fluxo de Ações para Verificar Eventos em Segundo Plano.	40
TABELA 17	–	Fluxo de Ações para Atualizar Agenda do Google no Dispositivo.	41

LISTA DE SIGLAS

API	Application Programming Interface
GPL	General Public License
GNU	GNU's Not Unix
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
URI	Uniform Resource Identifier
UML	Unified Modeling Language
IDE	Integrated Development Environment

SUMÁRIO

1	INTRODUÇÃO	9
2	OBJETIVOS	12
2.1	OBJETIVO GERAL	12
2.2	OBJETIVOS ESPECÍFICOS	12
3	METODOLOGIA	13
3.1	MODELAGEM DO SOFTWARE	14
4	MATERIAIS	15
4.1	VISUAL PARADIGM	17
4.2	ANDROID STUDIO	17
4.3	GENYMOTION	17
4.4	MOQUPS	17
5	DESENVOLVIMENTO	18
5.1	ANÁLISE DE REQUISITOS	18
5.2	DIAGRAMA DE CLASSES DE PROJETO DO APLICATIVO MÓVEL	19
5.3	DIAGRAMA DE ENTIDADE RELACIONAMENTO	20
5.4	DIAGRAMAS DE CASOS DE USO	21
5.5	DESCRIÇÃO DE CASOS DE USO	23
5.5.1	Gerenciador de Eventos	23
5.5.2	Aplicativo de Integração com Agenda	35
5.5.2.1	Lista de Telas da Aplicação Móvel	35
5.5.3	Diagramas de Sequência de Atividades	41
6	RESULTADOS	50
7	CONSIDERAÇÕES FINAIS	55
	REFERÊNCIAS	56

1 INTRODUÇÃO

O acúmulo de atividades torna necessária a utilização de recursos que possibilitem agilizar a realização do planejamento das atividades a serem executadas (SANCHEZ, 2015). Planejar o que será realizado nos permite criar estratégias para organizar o tempo disponível e proporcionar qualidade de vida e tempo para relaxar. É importante a atenção ao se planejar para um evento, pois informações devem ser registradas para acompanhar com eficiência as tarefas que serão realizadas. Participar de um evento requer planejamento devido ao fato de que podem ocorrer mudanças na programação, as quais nem sempre são repassadas a todos os participantes em tempo hábil. Quando estas ocorrem, o planejamento necessita ser refeito, podendo causar conflito com outros compromissos e impactando diretamente nas atividades dos participantes.

A tecnologia surgiu para proporcionar facilidade em aspectos sociais e econômicos propiciando, com a Internet e aparelhos móveis, o planejamento de tarefas com maior facilidade. A nova geração de consumidores está cada vez mais conectada, tornando necessário que as informações cheguem de forma ágil e que sejam recuperadas com a mesma agilidade (SOFTWARE, 2016). Segundo pesquisa realizada pela GlobalWebIndex 2017, empresa especializada em pesquisas na Internet, 94% das pessoas entre 16 e 24 anos responderam que possuíam um dispositivo móvel. Para este elevado número de dispositivos móveis, um novo nicho de mercado surgiu, o de desenvolvimento de aplicações para estes dispositivos.

Este nicho tornou necessário que organizadores de eventos considerassem o público que utiliza estes dispositivos móveis durante a divulgação e planejamento de eventos. Para atender esta demanda a empresa Eventbase (EVENTBASE, 2017) oferece a criação e disponibilização de aplicativos para estes dispositivos, os quais facilitam o planejamento e organização dos participantes. No entanto, possui limitações em relação às funcionalidades disponíveis somente em versões pagas. A empresa Eventbride (EVENTBRIDE, 2017) oferece uma plataforma para divulgação e vendas de ingressos para eventos, disponibilizando dois aplicativos: um para o organizador e outro para o participante. Estes aplicativos são utilizados para realizar buscas e compras, além de gerenciar informações relativas ao evento.

As empresas anteriormente mencionadas que disponibilizam soluções de gerenciamento de evento foram avaliadas devido a similaridade com a escolhida e por possuírem aplicativos para dispositivos móveis que permitem acompanhar a programação dos eventos. No entanto, não são as únicas soluções para gerenciamento de eventos no mercado (ADVICE, 2017). Apesar de robustas, as alternativas avaliadas não possuem em suas listas de funcionalidades, ferramentas que integrem a programação dos eventos à agenda digital dos participantes. Isto implica na necessidade de verificar periodicamente se não houveram mudanças na programação do evento e atualizar o planejamento, caso ocorram.

Os aplicativos existentes para gerenciamento de eventos requerem que o participante acompanhe a programação em um aplicativo próprio, geralmente desenvolvido somente para este determinado evento. Isto torna necessário que sejam instalados diversos aplicativos, desperdiçando recursos nos dispositivos. Para quem utiliza agendas para se organizar, torna-se incômodo e pouco prático verificar periodicamente estes aplicativos para realizar o planejamento de suas atividades, permitindo que ocorram conflitos entre eventos sem que o participante tenha tempo hábil de reagendar seus compromissos.

Participantes de eventos utilizam, com frequência, agendas para realizar o planejamento e programação das sessões a serem acompanhadas, podendo estas agendas ser tanto convencionais quanto digitais. Entretanto, devido a praticidade, as agendas digitais estão tornando-se cada vez mais presentes. Todo dispositivo móvel possui pelo menos um aplicativo com esta finalidade, fazendo com que seja mais cômodo utilizar a tecnologia do que registrar em uma agenda convencional, a tradicional agenda de papel.

Agendas digitais possuem inúmeros benefícios em relação às agendas convencionais. Elas permitem o compartilhamento em grupo, publicação na *Internet*, criação de lembretes e notificações, verificação de conflitos de horários na associação com outras agendas e dentre outras funcionalidades (NEUSTAEDTER et al., 2009). Devido aos benefícios e à crescente utilização de dispositivos móveis, houve uma diminuição no uso de agendas tradicionais (papel).

Com o aumento do uso de dispositivos móveis e do aumento de dispositivos móveis no mercado surgiram aplicativos que possibilitam o agendamento de tarefas, com toda a praticidade de uma agenda digital, mas além da praticidade, empresas como o *Google*, que disponibiliza em todos os aparelhos da plataforma *Android* aplicativos de agenda, disponibilizaram APIs para manipulação dos eventos inseridos na agenda. Isto facilitou que outras aplicações gerenciem os dados da agenda e integrem os dados de aplicativos com outros serviços disponibilizados, sem que fosse necessário conhecer como a agenda é programada, somente os métodos da *API*.

Devido as facilidades que a tecnologia propiciou, verificou-se a necessidade de integrar

as informações de eventos com a agenda pessoal dos participantes. Além disso, percebeu-se a necessidade de criar notificações quando houverem alterações nos horários das sessões de eventos. Assim, torna-se possível integrar as atividades dos eventos diretamente com a agenda pessoal do usuário, simplificando suas tarefas de planejamento.

Com base nas necessidades levantadas acima, foi identificada a necessidade de um aplicativo que se comunica com a agenda digital do participante e sincronize os dados de eventos. Buscando dados padronizados de uma aplicação de gerenciamento de eventos, permitindo a comunicação com o aplicativo sem gerar inconsistência. A sincronização deve iniciar quando um interessado no evento sinalize que deseja seguir a programação do evento na aplicação de gerenciamento de eventos. A partir deste momento, o aplicativo móvel realizará a aquisição dos dados referentes a programação das sessões dos eventos sinalizados. Feito isso, os dados serão processados e inseridos na agenda do *Google* no dispositivo móvel. A escolha desta agenda para sincronização foi motivada pela maior compatibilidade com a plataforma utilizada para desenvolvimento, além de possuir uma excelente documentação. Periodicamente o aplicativo deverá realizar a verificação de alterações nos horários, caso ocorram as atividades na agenda deverão ser atualizadas e o usuário deverá ser notificado de que que necessita reagendar as sessões que foram alteradas.

Conforme a figura 1, são necessários 2 (dois) componentes: a aplicação de gerenciamento de eventos, é o ponto em que as informações são concentradas e que disponibiliza uma API pública para aquisição dos dados; e o aplicativo móvel, que consome a API *WEB* e atualiza a agenda digital do cliente. São importantes também os papéis dos participantes e dos organizadores de eventos, que consomem e administram, respectivamente, as informações dos eventos.

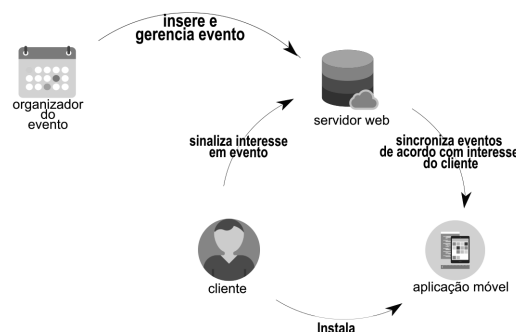


Figura 1: Modelo de funcionamento.

Fonte: O autor.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Implantar uma solução de gerenciamento de eventos e desenvolver um aplicativo para dispositivos móveis que realize a integração das atividades agendadas nos eventos com a agenda dos usuários.

2.2 OBJETIVOS ESPECÍFICOS

- Analisar projetos similares;
- Revisão de literatura sobre aplicativos móveis;
- Estudar o modelo RESTful e a API de comunicação da aplicação WEB e de integração com agenda;
- Realizar análise de requisitos e modelagem do aplicativo alvo;
- Desenvolver e testar protótipo para a plataforma Android;
- Ajustar aplicação web;
- Gerar documentação para o protótipo;

3 METODOLOGIA

Para a escolha do servidor de eventos para o aplicativo foi realizada a pesquisa por um projeto de código aberto, de forma que fosse possível modificar seus componentes para possibilitar a sincronização dos dados e acompanhar os horários das sessões sem custo para o usuário. A busca foi realizada no repositório de códigos-fonte *Github*(GITHUB, 2017) com o critério de aplicação para gerenciamento de eventos e que possuísse documentação e estivesse com desenvolvimento ativo e mantido pela comunidade, com uma licença livre.

Foi escolhido um projeto sob a licença GPL versão 3, pois, segundo Smith, "o *software* pode ser utilizado para qualquer propósito, pode ser modificado de acordo com as necessidades do usuário, ser distribuído para qualquer pessoa e ser compartilhado com quaisquer alterações". A licença GNU GPL versão 3 garante que projetos que a utilizam não sofram alterações futuras que restrinjam seu uso, permitindo que seja utilizado livremente, podendo servir como base para outros projetos sem que ocorra restrições quanto ao seu uso. Sendo assim, o projeto desenvolvido pela empresa *Fossasia* pode ser utilizado, modificado e distribuído sem que haja restrições impostas pela licença, desde que as quatro liberdades básicas do software livre sejam mantidas.

Foram realizadas as modificações necessárias no *software* de gerenciamento de eventos, para que fosse possível acompanhar a programação em um evento. Para sincronização com o aplicativo móvel foi utilizado o modelo *RESTful*, o qual utiliza os métodos do protocolo HTTP, sendo o formato de comunicação utilizado entre o servidor e o cliente o JSON. A *API* de comunicação já estava implementado, sendo necessário somente adicionar alguns métodos.

Foi desenvolvido um protótipo de aplicativo para a plataforma *Android*, este sincroniza os dados de eventos disponíveis em uma aplicação *web* e insere/atualiza entradas na agenda no dispositivo móvel do usuário que realizou a instalação do aplicativo. As informações sincronizadas dos eventos e sessões consistem no local, título, descrição e data e horário de cada sessão. Quando houver alterações nas datas ou horários das sessões, o aplicativo exibe uma notificação alertando o usuário e oferece a opção de remarcar em sua agenda caso tenha horário disponível.

Caso não haja horário disponível o aplicativo impede que seja marcado algo até que o evento conflitante seja removido ou alterado.

3.1 MODELAGEM DO SOFTWARE

Para o desenvolvimento deste trabalho foi realizado o estudo de requisitos para o aplicativo móvel e para o gerenciador de eventos, de forma a identificar o que era necessário implementar no aplicativo móvel e as alterações na aplicação de gerenciamento de eventos. O documento de requisitos foi desenvolvido na forma de tabela contendo os tipos e numeração dos requisitos, a descrição de cada requisito e seu grau de relevância para o projeto.

Foi realizada a modelagem e diagramação do *software* por meio dos diagramas da UML descritos a seguir. Para a elaboração dos diagramas foram utilizadas as diretrizes apresentadas por Booch.

- **Diagrama de Casos de Uso:** importante para visualizar, especificar e documentar ações realizadas no sistema analisado. Tornando-os acessíveis e compreensíveis, por apresentar uma visão externa sobre como esses elementos podem ser utilizados.
- **Diagrama de Classes:** permite a visualização das classes que compõem o sistema, com seus atributos e métodos, e visualizar os relacionamentos entre cada uma das classes, de forma estática, sem dar enfoque no fluxo temporal e ordem das mensagens.
- **Diagramas de Sequência:** especifica a ordenação temporal das mensagens no sistema, permitindo visualizar seu fluxo entre os diferentes objetos durante a execução das atividades do sistema.

Foi realizada a prototipação das telas do aplicativo móvel de forma a criar um modelo de como seria o aplicativo final. Os protótipos criados são apresentados juntamente com a especificação dos casos de uso.

4 MATERIAIS

Neste capítulo serão descritas as ferramentas utilizadas para o desenvolvimento do trabalho, tanto no servidor quanto no cliente. Para servidor foi selecionado o projeto *Open Event Organizer Server* (FOSSASIA, 2017), sob a licença GNU GPL versão 3, este projeto foi desenvolvido pela empresa *Fossasia* utilizando a linguagem de programação *Python* na versão 2.7 e o *framework* *Flask* (POCOO, 2017). A Figura 2 representa a página inicial da aplicação com as opções de criação de eventos e autenticação.

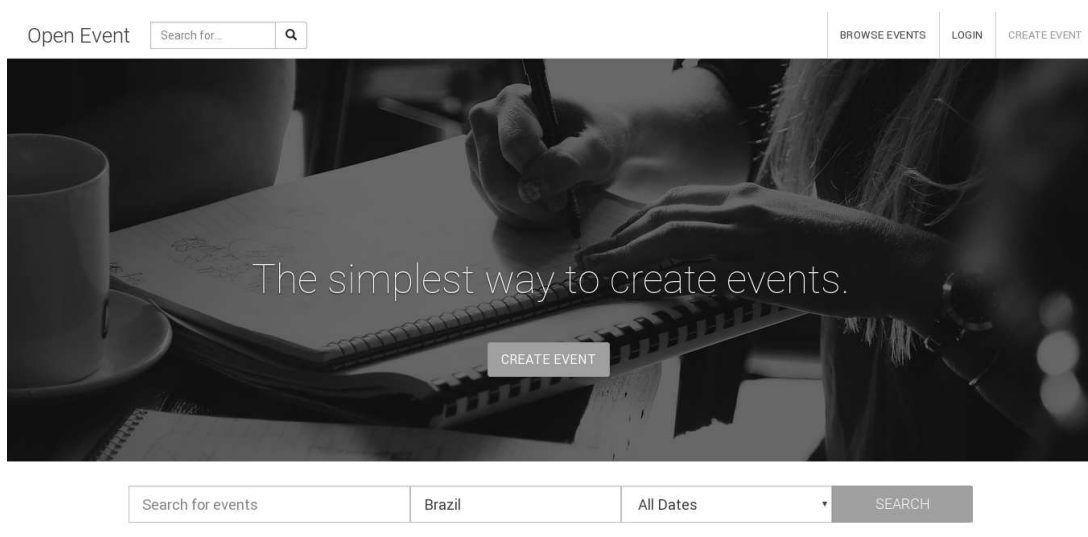


Figura 2: Página inicial da aplicação *Open Event Organizer Server*, segundo modelo padrão

Fonte: O autor

O projeto possui uma API seguindo o modelo REST, acessível por meio da *URL* do projeto e adicionando o caminho */api/v1/*. A página de início disponibiliza os métodos e URIs de cada recurso disponibilizado. A Figura 3 mostra a página inicial da *API* e os recursos oferecidos por ela.

Organizer Server APIs

Open Event Organizer APIs

events : Events	Show/Hide	List Operations	Expand Operations
sessions : Sessions	Show/Hide	List Operations	Expand Operations
tracks : Tracks	Show/Hide	List Operations	Expand Operations
speakers : Speakers	Show/Hide	List Operations	Expand Operations
sponsors : Sponsors	Show/Hide	List Operations	Expand Operations
microlocations : Microlocations	Show/Hide	List Operations	Expand Operations
login : Login	Show/Hide	List Operations	Expand Operations
users : Users	Show/Hide	List Operations	Expand Operations
notifications : Notifications	Show/Hide	List Operations	Expand Operations
attendees : Attendees	Show/Hide	List Operations	Expand Operations
tickets : Tickets	Show/Hide	List Operations	Expand Operations

[BASE URL: /api/v1 , API VERSION: 1.0]

Figura 3: Página com documentação da API utilizada

Fonte: O autor

Para a execução da aplicação de gerenciamento de eventos foi utilizada uma máquina com a distribuição GNU/Linux Debian versão 9.1 com codinome *Stretch* durante o procedimento de adaptação e implantada em uma instância na plataforma *Heroku* após as alterações serem concluídas. Os procedimentos para implantação na plataforma *Heroku* está disponível no repositório do projeto.

A plataforma *Heroku* (HEROKU, 2017) foi escolhida por permitir disponibilizar a aplicação de gerenciamento de eventos para *Internet* sem custos, apesar de restringir o tempo de execução. Para controle dos pacotes foi utilizada a ferramenta *virtualenv-wrapper* (HELLMANN, 2017) para que pacotes no sistema não interferissem na execução do projeto e que as versões dos pacotes utilizadas pelo desenvolvedor fossem mantidas. Para configuração e utilização da ferramenta foi seguido as instruções disponíveis na documentação oficial do pacote *virtualenv-wrapper*.

Para desenvolvimento do aplicativo foi utilizado a linguagem de programação nativa da plataforma Android, Java, tendo como versão mínima do sistema operacional o Android 4.4 com codinome *Kitkat*. Segundo pesquisa realizada pelo *Google* em dispositivos que utilizam a versão mais recente do *Google Play*, abrange 92.2% dos dispositivos existentes no mercado(GOOGLE, 2017).

4.1 VISUAL PARADIGM

O Visual Paradigm¹ é uma ferramenta que permite a modelagem de *software* utilizando a UML. Ela foi utilizada para a criação de diagramas de classe, de sequência e de casos de uso. O Visual Paradigm distribui duas versões para utilização, uma paga para uso empresarial e outra gratuita, foi escolhido a versão gratuita, pois suas funcionalidades foram suficientes para desenvolver os diagramas pretendidos.

4.2 ANDROID STUDIO

O Android Studio² é um ambiente de desenvolvimento integrado (IDE) e foi utilizado para desenvolvimento da aplicação móvel. Esta é a ferramenta oficial para desenvolvimento para a plataforma Android, possui facilidades que auxiliam no desenvolvimento e modelagem do aplicativo, tais como verificação de erros de compilação e depuração.

4.3 GENYMOTION

O Genymotion³ é um *software* que permite que seja emulado um sistema operacional Android em um computador que tenha os sistemas operacionais *Linux*, *Windows* ou *Mac OS*. Ele permite que seja emulado diversos modelos de dispositivos, tendo sido utilizado para testar a aplicação em desenvolvimento.

4.4 MOQUPS

A ferramenta Moqups⁴ permite que sejam desenvolvidos protótipos visuais e de interação com a interface de usuário e foi utilizada para modelar as telas da aplicação móvel. A ferramenta possui limitações quanto ao uso de forma gratuita, mas atendeu aos requisitos necessários para desenvolvimento do projeto.

¹<https://www.visual-paradigm.com/> - Página Inicial da ferramenta Visual Paradigm

²<https://developer.android.com/studio/index.html?hl=pt-br> - Página contendo instrução para instalação da *IDE*

³<https://www.genymotion.com/> - Página inicial do emulador Genymotion

⁴<https://moqups.com/> - Página inicial da ferramenta Moqups

5 DESENVOLVIMENTO

5.1 ANÁLISE DE REQUISITOS

Tipo	Descrição	Grau de Importância
RF01	Permitir o gerenciamento de Eventos por meio da interface <i>web</i> para organizadores de eventos	Alto
RF02	Permitir seguir um evento por meio de uma interface <i>web</i>	Alto
RF03	Autenticar usuário em aplicação móvel	Alto
RF04	Inserir a programação dos eventos inscritos na agenda móvel de usuários	Alto
RF05	Notificar usuário de alterações na programação de evento	Alto
RF06	Permitir a seleção de itens na programação de evento	Alto
RF07	Permitir a criação de uma nova agenda para salvar os eventos	Baixo
RF08	Listar eventos que o usuário irá participar através do aplicativo móvel	Alto
RF09	Permitir a submissão de propostas de sessões para o evento	Moderado
RF10	Gerenciar sessões do evento	Alto
RF11	Gerenciar horários do evento	Alto
RF12	Permitir o cadastro de usuários na aplicação <i>web</i>	Alto
RF13	Usuário deve poder sincronizar manualmente os dados dos eventos.	Moderado
RNF01	Verificar se os horários de itens de um evento conflitam com atividades previamente agendadas	Baixo
RNF02	As informações de programação e eventos que o usuário irá participar deverão ser sincronizadas automaticamente	Alto

RNF03	O usuário deverá informar as suas credenciais para sincronizar as informações de eventos	Alto
RNF04	Eventos não deverão ser inseridos na agenda até que o usuário selecione itens que deseja participar	Alto
RNF05	Cada evento deve conter informações de status em relação ao agendamento das palestras	Alto
RNF06	Cada palestra deve conter as informações de status na agenda (agendado, ignorado)	Alto
RNF07	Alterações na programação do evento deverão ser notificadas ao usuário, fazendo com que ele tenha que aceitar ou não a mudança em sua agenda. Alterando o status da sessão para pendente.	Alto
RNF08	Dados cadastrais de Evento devem ser validados	Alto
RNF09	Quantidade de ingressos para evento deve ser especificada	Moderado
RNF10	Sessão na agenda do dispositivo deverá conter o nome do evento, título da sessão e orador em sua descrição	Alto

Tabela 1: Análise de Requisitos

5.2 DIAGRAMA DE CLASSES DE PROJETO DO APLICATIVO MÓVEL

O aplicativo móvel mantém somente as informações necessárias do evento e dos horários das sessões para visualização e integração com a agenda do usuário, as demais informações não são salvas. Ficando armazenadas na aplicação de gerenciamento de eventos.

A figura 4 representa o diagrama que foi realizado para documentação das classes implementadas no projeto e suas relações umas com as outras, o diagrama é dividido em seis pacotes que contém as principais classes do projeto: o pacote *models* contém as classes que são utilizadas para mapeamento das entidades no banco de dados utilizado; o pacote *services* e *receivers* contém as classes utilizadas para execução de serviços em segundo plano, como atualização e notificação de alteração das sessões de eventos; o pacote *requests* contém as classes necessárias para sincronização dos eventos e as principais regras de funcionamento da aplicação; o pacote *activities* contém as classes principais do *framework Android* para exibição de componentes gráficos, interação e tratamento de eventos do usuário; o pacote *adapters* contém as classes responsáveis pela exibição dos eventos e sessões para o usuário.

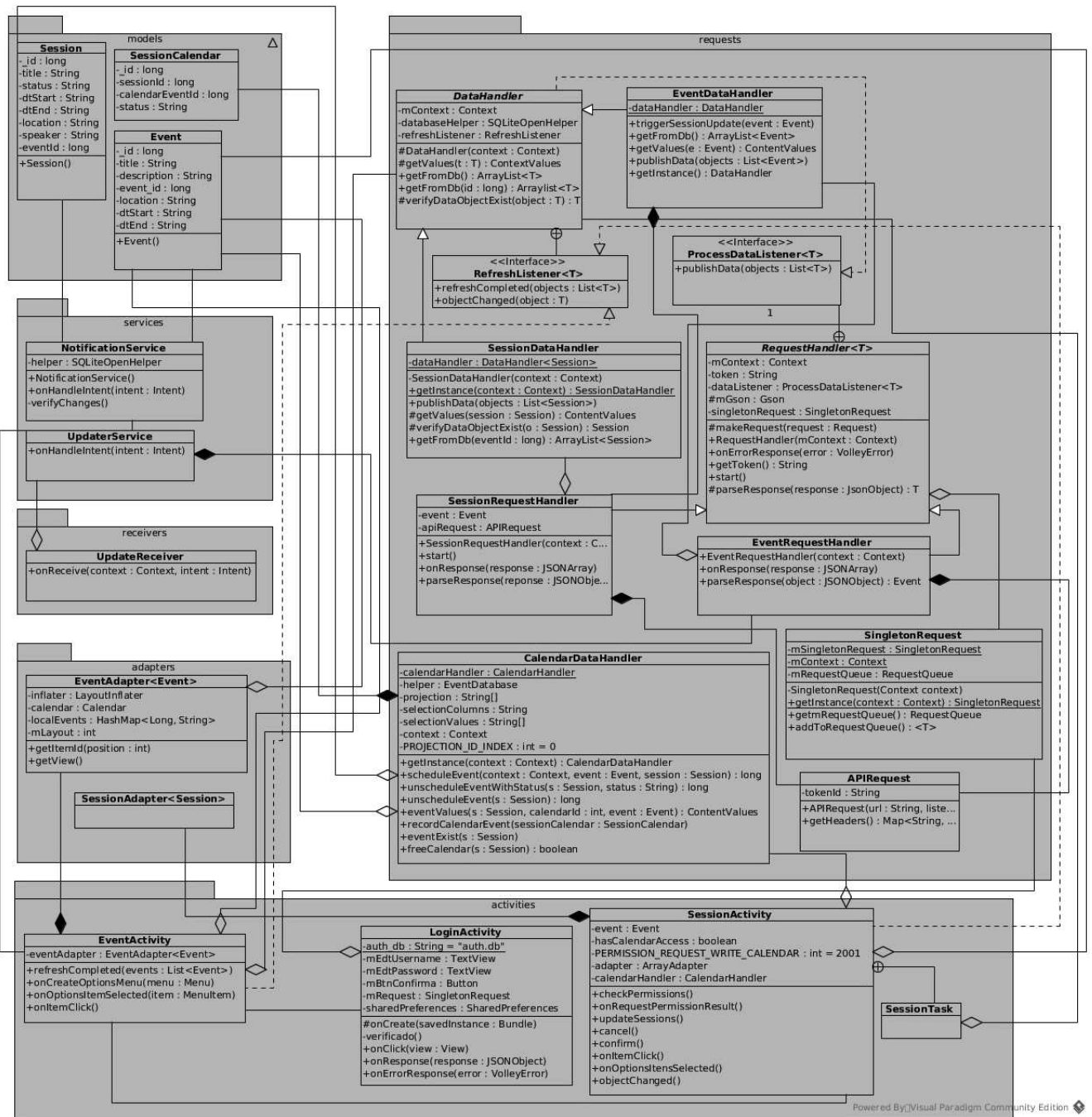


Figura 4: Diagrama de Classes do Aplicativo Desenvolvido.

Fonte: O autor.

5.3 DIAGRAMA DE ENTIDADE RELACIONAMENTO

Foram especificadas as tabelas utilizadas para armazenar as informações no banco de dados do dispositivo. Na figura 5 estão as três tabelas utilizadas para salvar informações na aplicação e suas relações. A API utilizada permite que seja obtido um conjunto amplo de

informações, no entanto, somente as informações pertinentes foram mantidas no banco de dados do aplicativo.

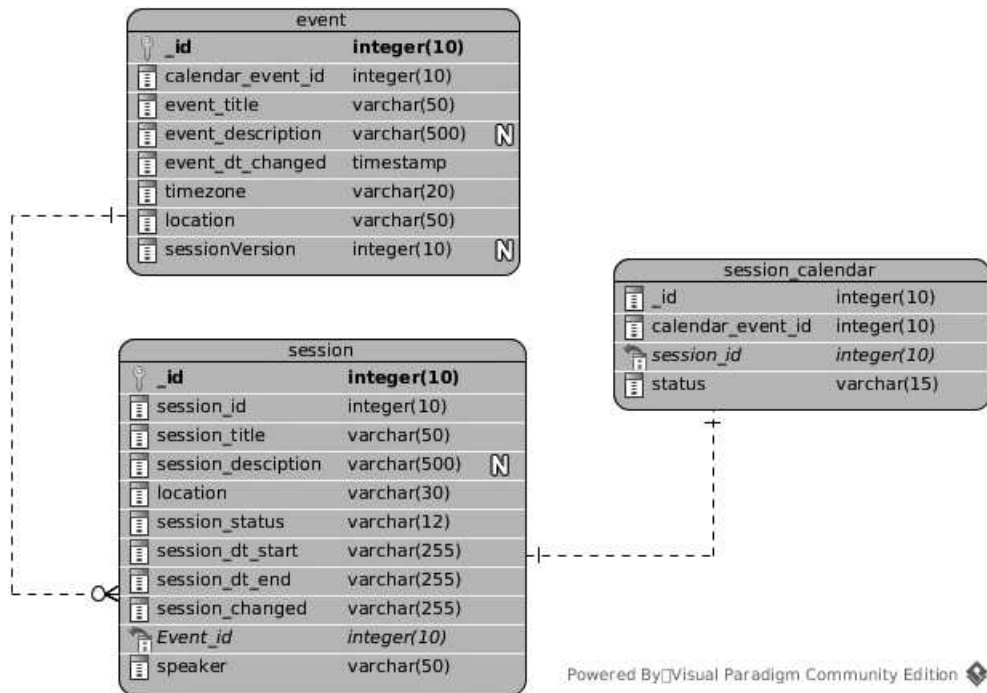


Figura 5: Diagrama de Entidade Relacionamento do Banco de Dados do aplicativo desenvolvido.

Fonte: O autor.

5.4 DIAGRAMAS DE CASOS DE USO

Devido a utilização de uma aplicação existente para gerenciamento de eventos, foram destacados em laranja na figura 6 os casos de uso necessários para entendimento da utilização da aplicação e do que foi alterado. Na figura 7 foram destacados os casos de uso do aplicativo que foi desenvolvido, para entendimento das funções que cada ator no sistema desempenha.

Os casos de uso destacados, devido a complexidade e importância para o projeto, foram especificados de forma detalhada. Devido a complexidade da aplicação de gerenciamento de evento e de funções que não são pertinentes ao desenvolvimento do projeto, foi escolhido omitir casos de uso no diagrama para destacar os principais.

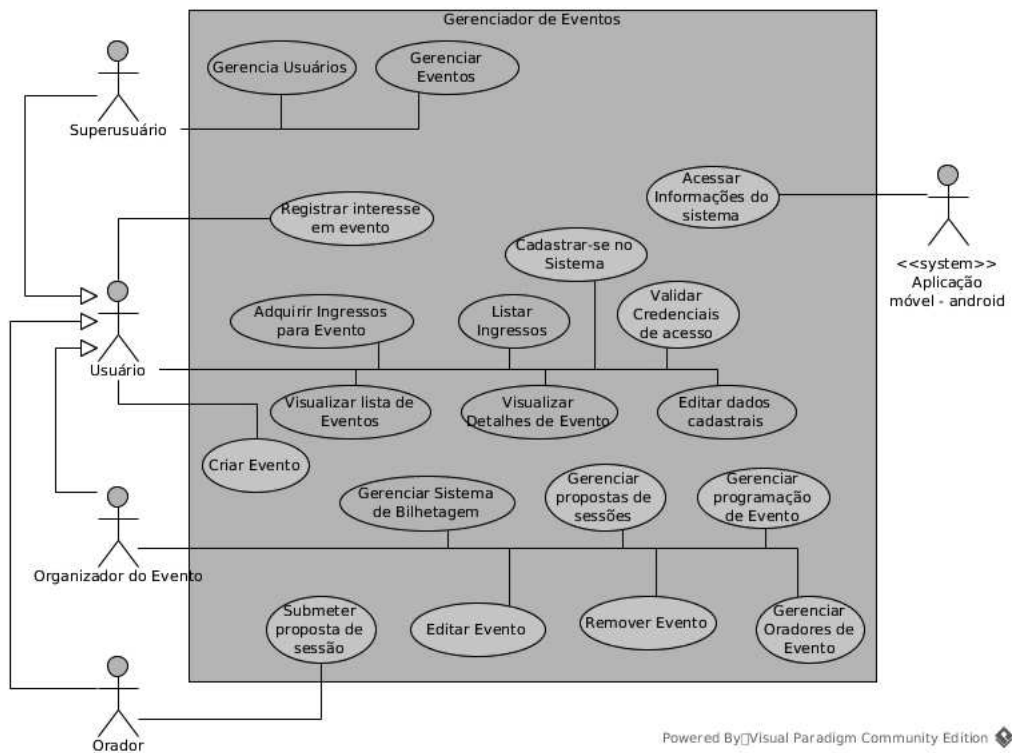


Figura 6: Diagrama de Casos de Uso da Aplicação de Gerenciamento de Eventos

Fonte: O autor

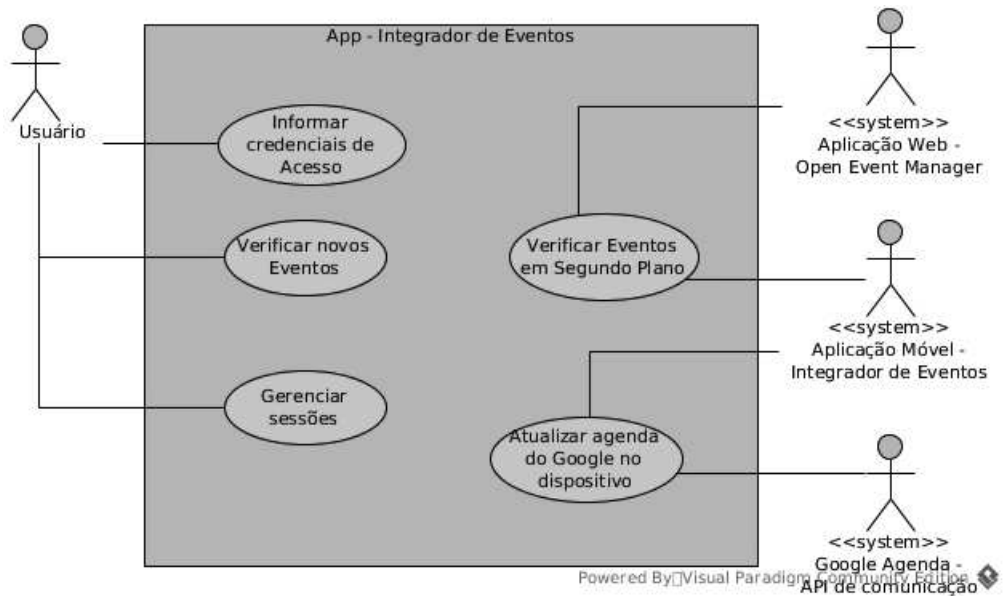


Figura 7: Diagrama de Casos de Uso do Aplicativo Móvel

Fonte: O autor

5.5 DESCRIÇÃO DE CASOS DE USO

5.5.1 GERENCIADOR DE EVENTOS

Aplicação que permite que seja cadastrado e gerenciado eventos. Possui um sistema de gerenciamento de horários de sessões e uma *API*, que permite acesso aos dados dos eventos cadastrados no sistema. A seguir foi realizado a especificação dos casos de uso da aplicação de gerenciamento de eventos, mostrando a interação entre os atores e seus determinados papéis no sistema.

Caso de Uso: Criar Evento	
Requisitos	RF01 RN08
Descrição	Permite a criação de um evento no sistema de forma que possa ser exibido publicamente no gerenciador de eventos.
Atores	Usuário.
Pré-condições	Usuário possuir credenciais válidas no sistema
Pós-condições	Evento criado ou definido como rascunho
Fluxo Básico	
Usuário	Sistema
1. Acessa URL para criação de evento	2. Redireciona para tela de criação de evento
3. Informa dados obrigatórios (nome do evento, localização, data e hora de início e término) e dados opcionais. Clica em salvar e prosseguir.	4. Salva informações do evento e redireciona para próxima tela. Validando informações obrigatórias, permanece na página caso alguma informação obrigatória não estiver inserida
5. Insere informações de patrocinadores e clica no botão de salvar e prosseguir.	6. Salva informações de patrocinadores e redireciona para a próxima tela. Não é necessário validação nos campos
7. Insere informações de sessões e oradores. Clica no botão de salvar e prosseguir.	8. Salva dados e redireciona para tela de administração de evento.
Exceções	
4. Informações obrigatórias não informadas: Sistema redireciona para passo 3.	
Fluxo Alternativo: habilita sistema de bilhetagem	

3. Clica no botão de habilitar sistema de bilhetagem	4. Exibe campos de informações para bilhetagem
5. Insere informações necessárias para execução do sistema de bilhetagem. É necessário escolher o tipo de ingresso, informando a quantidade de cada um. Para ingressos pagos deve ser inserido instruções para pagamento.	6. Executa passo 4 do fluxo básico, para salvar informações do evento.
Fluxo Alternativo: habilita informações do	
3. Clica no botão de habilitar informações do organizador	4. Exibe campos de informações para bilhetagem
5. Insere informações do organizador.	6. Executa passo 4 do fluxo básico. Não valida informações.
Fluxo Alternativo: Publica evento	
3,5,7 Clica no botão de publicar evento.	Sistema salva dados e redireciona para tela de administração de evento. Termina caso de uso.
Fluxo Alternativo: Desativar informações de patrocinadores	
5. Clica no botão de desativar patrocinador	6. Desativa informações de patrocinadores e redireciona para próxima tela
Fluxo Alternativo: Desativar informações de sessões e oradores	
7. Clica no botão de desativar informações de oradores e sessões.	8. Desativa informações de oradores e sessões e redireciona para tela de administração de evento.

Tabela 2: Fluxo de Ações para Criar Evento.

Caso de Uso: Editar Evento	
Descrição	O fluxo de edição é o mesmo do Caso de Uso: Criar Evento. Caso o usuário não tenha publicado o evento, o sistema permite que realize a publicação. Caso o evento esteja publicado, o usuário poderá desfazer a publicação a qualquer momento, na edição.
Requisitos	Complementa o requisito RN01.

Atores	Organizador de Evento
Pré-condições	<ul style="list-style-type: none"> • Usuário possuir credenciais válidas no sistema • Evento criado • Permissões para edição de evento • Usuário estar na tela gerenciamento do evento
Pós-condições	<ul style="list-style-type: none"> • Informações do evento alteradas.
Fluxo Básico	
Organizador de Evento	Sistema
1. Clica no botão de editar evento	2. Redireciona para tela de edição do evento.
3. Demais passos iguais a Caso de Uso: Criar Evento	

Tabela 3: Fluxo de Ações para Editar Evento.

Caso de Uso: Remover Evento	
Descrição	Remove evento da lista de eventos públicos e da lista de eventos gerenciados pelo usuário.
Requisitos	Evento deve ser mantido para visualização por superusuário. Complementa o requisito RF01.
Atores	Organizador de Evento
Pré-condições	<ul style="list-style-type: none"> • Evento criado • Permissão para remover evento. • Estar na tela de gerenciamento de evento
Pós-condições	<ul style="list-style-type: none"> • Evento removido da lista de eventos gerenciados pelo usuário.
Fluxo Básico	
Organizador de Eventos	Sistema

1. Clica no botão de remover evento	2. Exibe mensagem de confirmação e campo de texto para digitar nome do evento.
3. Insere nome do evento e clica no botão de confirmar exclusão.	4. Valida nome do evento e o remove. Redireciona para tela que exibe todos os eventos gerenciados por usuário.
Exceções	
3. Clica no botão de cancelar: Encerra caso de uso. 4. Nome incorreto: volta para passo 3	

Tabela 4: Fluxo de Ações para Remover Evento.

Caso de Uso: Gerenciar Propostas de Sessão para evento	
Descrição	Permite que seja gerenciado as propostas enviadas por oradores para sessões do evento. As sessões possuem um status, para cada status um conjunto diferente de ações pode ser realizado. A tela de gerenciamento de sessões exibe informações resumidas e uma coluna de ações a serem realizadas para cada sessão.
Requisitos	<ul style="list-style-type: none"> • A tela de gerenciamento de sessão deve possuir um atalho que redirecione para a tela de criação de proposta de sessão. • Satisfaz RN14
Atores	Organizador de Eventos
Pré-condições	<ul style="list-style-type: none"> • Usuário logado no sistema • Permissão para gerenciar sessões • O evento deve estar criado e com o sistema de sessões habilitado. • Permitir a submissão de propostas de sessões para o evento

Pós-condições	<ul style="list-style-type: none"> • Status de sessão alterado • Sessão criada • Sessão editada
Fluxo Básico	
Organizador de Eventos	Sistema
1. Clica no botão de alteração de status da sessão desejado e envia e-mail de confirmação de aceite de proposta.	2. Exibe tela de envio de e-mail, contendo assunto e mensagem de .
3. Clica em enviar e-mail.	4. Envia e-mail e altera sessão para status desejado.
Exceções	
1. Não clica em nada: Termina caso de uso.	
1. Não existe proposta de sessão disponível: Permite a criação de proposta de sessão.	
3. Clica em cancelar: Termina caso de uso.	
Fluxo Alternativo: Editar Sessão	
1. Clica no botão de editar sessão.	2. Redireciona para tela de edição de sessão
3. Demais passos igual a caso de uso: Submeter Proposta de Sessão	
Fluxo Alternativo: Remover Sessão	
1. Clica no botão de remover sessão.	2. Exibe mensagem para confirmação de exclusão.
3. Confirma exclusão de sessão.	4. Remove sessão de evento.
Exceção	
3. Cancela exclusão: Finaliza Fluxo.	

Tabela 5: Fluxo de Ações para gerenciar propostas de sessões para evento.

Caso de Uso: Gerenciar Oradores de Evento	
Descrição	Permite a visualização da lista de oradores. Exibindo informações básicas de cada e informações como título e status relativas das sessões propostas por orador. Permite que seja alterado informações de cadastro do orador e que seja removido orador do evento.
Requisitos	<ul style="list-style-type: none"> • Remover orador deve excluir todas as sessões associadas a ele. • Tela de gerenciamento de oradores deve possuir um link que redireciona para tela de submissão de evento, adicionando um orador.
Atores	Organizador de Eventos
Pré-condições	<ul style="list-style-type: none"> • Evento criado. • Opção de sessões e oradores habilitado. • Estar na tela de gerenciamento de Oradores.
Pós-condições	<ul style="list-style-type: none"> • Orador com dados alterados. • Orador excluído.
Fluxo Básico	
Organizador de Eventos	Sistema
1. Seleciona filtro de oradores por status da sessão.	2. Retorna lista de oradores conforme filtro selecionado.
3. Clica no botão de Visualizar Sessão	4. Exibe tela com as informações inseridas relativa a sessão selecionada.
Exceção	
3. Não clica em nada: Termina caso de uso.	
Fluxo Alternativo: Remove Sessão	
3. Clica no botão de remover sessão	4. Exibe tela de confirmação de remoção.

5. Confirma exclusão.	6. Exclui sessão e retorna para tela de gerenciamento de sessão.
Exceção	
5. Cancela exclusão: Termina Fluxo.	
Fluxo Alternativo: Editar informações de Orador	
3. Clica no botão de editar informações do orador	4. Redireciona para tela de edição de informações.
5. Preenche dados obrigatórios (nome e e-mail) e informações adicionais. Salva dados inseridos.	6. Valida informações digitadas e redireciona para tela de gerenciamento de oradores.
Exceções	
5. Cancela edição: Redireciona para tela de gerenciamento de oradores	
6. Dados inválidos: Volta para tela de edição de informações de orador.	

Tabela 6: Fluxo de Ações para Gerenciar oradores de sessões.

Caso de Uso: Gerenciar Programação de Evento	
Descrição	As sessões que foram aprovadas e confirmadas na programação do evento ficam disponíveis para inserção na grade de horários do evento, que consiste em uma tabela contendo em cada coluna uma das microrregiões do evento. As sessões são distribuídas manualmente pelo organizador do evento.
Requisitos	<ul style="list-style-type: none"> ● A programação do evento pode ser publicada ou removida a qualquer instante. ● As sessões não inseridas no cronograma ficarão destacadas. ● Não deverá ser possível inserir duas sessões no mesmo horário e microrregião. ● Satisfaz RF11.

Atores	Organizador de Eventos
Pré-condições	<ul style="list-style-type: none"> • Sessões aprovadas/confirmadas • Permissão para gerenciar evento • Estar na tela de gerenciamento de horários
Pós-condições	<ul style="list-style-type: none"> • Cronograma alterado.
Fluxo Básico	
Organizador de Eventos	Sistema
1. Clica e arrasta sessão para dia e horário desejado na tabela de horários.	2. Salva alterações e exibe notificação de que ação foi completada.
Fluxo Alternativo: Remove sessão da programação	
1. Clica no botão de remover da tabela de horários a sessão alvo.	2. Remove sessão da tabela de horários e altera status da sessão para não programada.

Tabela 7: Fluxo de Ações para Gerenciar Programação de Evento.

Caso de Uso: Cadastrar-se no sistema	
Descrição	Permite que usuário tenha credenciais válidas no sistema, podendo acessar recursos do sistema como comprar ingressos, criar e gerenciar eventos, submeter sessões e acompanhar eventos.
Requisitos	Satisfaz RF12.
Atores	Usuário
Pré-condições	<ul style="list-style-type: none"> • Email não cadastrado no sistema.
Pós-condições	<ul style="list-style-type: none"> • Usuário criado
Fluxo Básico	
Usuário	Sistema

1. Insere e-mail, senha e confirmação da senha	2. Valida e-mail na base de dados, tamanho mínimo da senha e cria usuário. Autentica usuário e redireciona para tela inicial do evento.
Exceções	
2. Usuário já possui e-mail cadastrado ou comprimento da senha menor que 6: volta ao passo um, informando mensagem de erro.	

Tabela 8: Fluxo de Ações para se Cadastrar no Sistema.

Caso de Uso: Validar Credenciais de Acesso	
Descrição	Valida usuário na página do sistema com base no usuário e senha informado.
Requisitos	Complementa RF12
Atores	Usuário
Pré-condições	<ul style="list-style-type: none"> • Possuir credenciais válidas no sistema.
Pós-condições	<ul style="list-style-type: none"> • Acesso a páginas restritas.
Fluxo Básico	
Usuário	Sistema
1. Insere e-mail e senha.	2. Valida e-mail e senha na base de dados. Autentica usuário e redireciona para tela inicial do evento.
Exceções	
2. E-mail ou senha não correspondem aos cadastrados no sistema: retorna para passo 1	

Tabela 9: Fluxo de Ações para Validar Credenciais de Acesso.

Caso de Uso: Registrar Interesse em Evento	
Descrição	Usuário registra que deseja acompanhar o evento, que consiste em receber notificações sobre a programação e alterações nos horários de sessões através de aplicação para Android, é independente de comprar ingressos ou não.
Requisitos	Implementado RF12
Atores	Usuário
Pré-condições	<ul style="list-style-type: none"> • Usuário logado
Pós-condições	<ul style="list-style-type: none"> • Registrado interesse em acompanhar evento
Fluxo Básico	
Usuário	Sistema
1. Usuário acessa página de detalhes de evento. Clica no botão de Seguir Evento	2. Salva preferência de usuário e exibe mensagem com confirmação de realização da operação.
Exceções	
2. Usuário não está logado no sistema. Redireciona para tela de login do sistema.	

Tabela 10: Fluxo de Ações para registrar interesse em evento.

Caso de Uso: Submeter Proposta de Sessão	
Descrição	Submete proposta de sessão para apresentação em evento dentro do período cadastrado.
Requisitos	<ul style="list-style-type: none"> • Usuário deve inserir informações para serem utilizadas em sessão. Requerido a inserção de nome por padrão, demais campos obrigatórios são configurados na tela edição do evento.
Atores	Usuário

Pré-condições	<ul style="list-style-type: none"> • Usuário logado • Submissões de proposta de sessão aberta para envio.
Pós-condições	<ul style="list-style-type: none"> • Sessão submetida para organizador de evento
Fluxo Básico	
Usuário	Sistema
1. Clica no botão de adicionar proposta de sessão	2. Redireciona para tela de inserção de sessão.
3. Insere campos obrigatórios (título de sessão e tipo de sessão). Demais campos opcionais. Clica no botão de salvar sessão.	4. Valida campos obrigatórios. Salva sessão e redireciona para tela de sessões de usuário.
Exceções	
4. Campos obrigatórios não informados: Volta para passo 3.	

Tabela 11: Fluxo de Ações para Submeter Proposta de Sessão.

Caso de Uso: Acessar Informações do Sistema	
Descrição	Permite que informações inseridas na base de dados sejam acessadas por aplicações de terceiros utilizando uma API disponibilizada publicamente. Informações relativas a programação, dados do evento, de participantes, oradores e etc.

Requisitos	<ul style="list-style-type: none"> • Métodos que exigem alterações ou que expõem informações pessoais devem utilizar as credenciais de usuário para autorizar operação. • URL para conexão utilizada deve conter o endereço da aplicação com o caminho “/api/v1/” • Credenciais para autorização devem ser obtidas através de token gerado pela aplicação.
Atores	<ul style="list-style-type: none"> • Aplicação Móvel – Android. • Aplicação Web - Open Event Manager.
Pré-condições	<ul style="list-style-type: none"> • Usuário possuir permissões para acessar conteúdo que deseja. • API para acesso a dados implementada.
Pós-condições	<ul style="list-style-type: none"> • Informações requeridas obtidas.
Fluxo Básico	
Aplicação Móvel – Android	Aplicação Web - Open Event Manager
1. Envia requisição para acessar informações da <i>API</i> .	2. Sistema valida e processa requisição. Retorna dados requeridos por usuário.
Exceções	
<p>2. Credenciais inválidas ou não informadas para método protegido: Sistema retorna erro de credenciais inválidas.</p> <p>2. Requisição a método não implementado ou inexistente: Sistema retorna que não possui esta função implementada.</p>	

Tabela 12: Fluxo de Ações para Acessar Informações do Sistema.

5.5.2 APLICATIVO DE INTEGRAÇÃO COM AGENDA

Permite que seja sincronizado a programação de eventos que usuário deseja participar em sua agenda pessoal. Acompanhando assim a programação e as mudanças que ocorrem ao decorrer do evento. A seguir está a descrição dos casos de uso identificados para a aplicação que realiza a integração com a agenda.

5.5.2.1 LISTA DE TELAS DA APLICAÇÃO MÓVEL



Figura 8: Representação de Tela de Login.

Fonte: O autor.



Figura 9: Tela de Visualização de eventos.

Fonte: O autor.

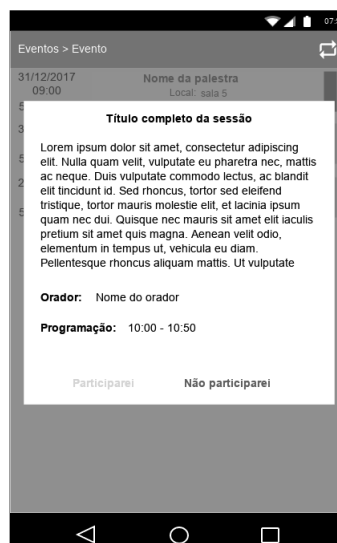


Figura 10: Tela de Gerenciamento de Sessões.

Fonte: O autor.

Caso de Uso: Informar Credenciais de Acesso	
Tela Esperada	Figura 8
Descrição	Permite que usuário acesse aplicativo utilizando as credenciais cadastradas em aplicação web.
Requisitos	<ul style="list-style-type: none"> • Caso de uso atende a requisito RF03. • Oferecer opção de salvar credenciais de acesso. • Solicitar credenciais somente no primeiro acesso. • A partir do momento que o aplicativo validar as credenciais o sistema deverá sincronizar com o aplicativo os eventos e horários das sessões que o usuário se inscreveu.
Atores	Usuário
Pré-condições	<ul style="list-style-type: none"> • Aplicativo iniciado • Possuir credenciais válidas na aplicação web.
Pós-condições	<ul style="list-style-type: none"> • Credenciais validadas • Usuário redireciona para página de visualização de eventos.
Fluxo Básico	
Usuário	Sistema
1. Usuário insere credenciais de acesso (e-mail, senha). Clica no botão de Realizar autenticação.	2. Sistema valida se e-mail está no formato correto e encaminha requisição para aplicação web que valida com credenciais cadastradas. Redireciona para página de visualização de eventos.
Exceções	
2. E-mail em formato incorreto: Informa usuário e retorna para passo 1	

2. Credenciais não correspondem com as cadastradas em aplicação web: Informa usuário e volta para passo 1.	
Fluxo Alternativo: Não possui credenciais de acesso	
1. Usuário não possui credenciais de acesso e clica em botão de inscrever-se no sistema.	2. Sistema redireciona para tela de cadastro de usuário

Tabela 13: Fluxo de Ações para Informar Credenciais de Acesso no Aplicativo Móvel.

Caso de Uso: Verificar novos Eventos	
Telas Esperadas	<ul style="list-style-type: none"> • Figura 9 • Figura 10
Descrição	Permite que usuário realize a verificação de um novo evento manualmente por meio da página de visualização de eventos.
Requisitos	<ul style="list-style-type: none"> • Satisfaz RF17;
Atores	Usuário
Pré-condições	<ul style="list-style-type: none"> • Usuário logado no aplicativo. • Aplicativo aberto na tela de visualização de eventos. • Implementar verificação de versão de horários das sessões.
Pós-condições	<ul style="list-style-type: none"> • Componente de lista de Eventos Atualizada. • Banco de dados interno atualizado.
Fluxo Básico	
Usuário	Sistema
1. Clica no botão de atualizar eventos (figura 9).	2. Consulta servidor web para visualizar novas alterações na programação. Atualiza componente que contém lista de evento e salva no banco de dados interno da aplicação.

Exceções
2. Nenhuma alteração em programação: Termina caso de uso.

Tabela 14: Fluxo de Ações para Verificar Novos Eventos.

Caso de Uso: Gerenciar Sessões	
Descrição	Permite ao usuário adicionar ou remover sessões do evento à sua agenda pessoal do Google.
Requisitos	<ul style="list-style-type: none"> • Status de evento deve ser alterado com a ação de remover ou adicionar evento à agenda do Google; • Deve ser verificado conflitos de horários entre sessões e eventos previamente programados; • Sessões pendentes deverão aparecer com cor diferenciada; • O usuário poderá definir dois status: agendado ou ignorado; • O status padrão deverá ser pendente, sendo possível somente alterar para o status definido pelo usuário; • Sessão deve ser inserida ou removida de Agenda do Google; • Satisfaz RF04;
Atores	Usuário
Pré-condições	<ul style="list-style-type: none"> • Sincronização de Eventos implementada • Usuário logado • Usuário na tela de visualização de sessões de Evento

Pós-condições	<ul style="list-style-type: none"> • Sessão inserida/removida de Agenda do dispositivo
Fluxo Básico	
Usuário	Sistema
1. Usuário clica em sessão do evento.	2. Sistema exibe a tela de confirmação de inserção na agenda caso a sessão não tinha sido agenda. Caso contrário exibe tela de confirmação de remoção da agenda.
3. Usuário seleciona opção de confirmar.	4. Executa Operação.
Exceção	
3. Usuário não confirma operação: Caso de uso encerrado.	

Tabela 15: Fluxo de Ações para Gerenciar Sessões.

Caso de Uso: Verificar Eventos em Segundo Plano	
Descrição	Permite sincronizar eventos que usuário seguir na aplicação web com a aplicativo em seu <i>smartphone</i> e que alterações nestes eventos sejam verificadas e notificadas.
Requisitos	<ul style="list-style-type: none"> • Satisfaz RN07; • Satisfaz RN02; • Satisfaz RF05; • Satisfaz RN01;
Atores	<ul style="list-style-type: none"> • Aplicação Móvel – Integrador de Eventos; • Aplicação Web – Open Event Manager;

Pré-condições	<ul style="list-style-type: none"> • Usuário logado e com credenciais válidas na aplicação web; • Usuário com conta válida na aplicação web.
Pós-condições	<ul style="list-style-type: none"> • Caso houverem alterações nas sessões, usuário notificado alterações;
Fluxo Básico	
Aplicação Móvel – Integrador de Eventos	Aplicação Web – Open Event Manager
1. Inicia sincronização de com Aplicação Web através de URL inserida em arquivo de configuração.	2. Valida credenciais de acesso de usuário e retorna informações de todos os eventos seguidos pelo usuário.
3. Processa informações obtidas. 4. Notifica usuário caso houverem alterações nos horários de sessões e salva alterações no banco de dados, se versão de programação for alterada. 5. Exibe na tela de sessões novas entradas caso houverem.	
Exceções	
2. Credenciais de acesso inválidas: Retorna erro de falta de permissão para acesso;	
4. Evento sem alterações: termina caso de uso.	

Tabela 16: Fluxo de Ações para Verificar Eventos em Segundo Plano.

Caso de Uso: Atualizar agenda do <i>Google</i> no dispositivo	
Descrição	Inserir ou remover sessão de evento na agenda do <i>Google</i> . Mostrando com um status diferente quando um evento estiver presente ou não na agenda do <i>Google</i> .
Requisitos	<ul style="list-style-type: none"> • Satisfaz RN10;

Atores	<ul style="list-style-type: none"> • Aplicação Móvel – Integrador de Eventos; • Aplicação Móvel – Integrador de Eventos;
Pré-condições	<ul style="list-style-type: none"> • Possuir uma agenda no dispositivo • Possuir sessões a serem inseridas na agenda
Pós-condições	<ul style="list-style-type: none"> • Sessão inserida/atualizada na agenda; • Atualizado status de sessão no aplicativo;
Fluxo Básico	
Aplicação Móvel – Integrador de Eventos	Aplicação Móvel – Integrador de Eventos
1. Insere/remove na agenda do dispositivo sessão que usuário participará ou que não participará.	2. Salva informações na agenda e retorna código de operação ou identificador de evento inserido.
3. Atualiza banco de dados interno com novo status de sessão.	
Exceção	
1. Sessão já marcada por usuário.	

Tabela 17: Fluxo de Ações para Atualizar Agenda do Google no Dispositivo.

5.5.3 DIAGRAMAS DE SEQUÊNCIA DE ATIVIDADES

A seguir estão os diagramas de sequência especificados para documentar as mensagens trocadas entre as classes do projeto, para entendimento de como as classes interagem entre si e como uma dependem uma da outra.

A figura 11 especifica a sequência de mensagens trocadas ao executar o caso de uso de gerenciar sessões pelo usuário. Descrevendo as mensagens trocadas entre as classes envolvidas na ação, de acordo com o caso de uso de gerenciar sessões.

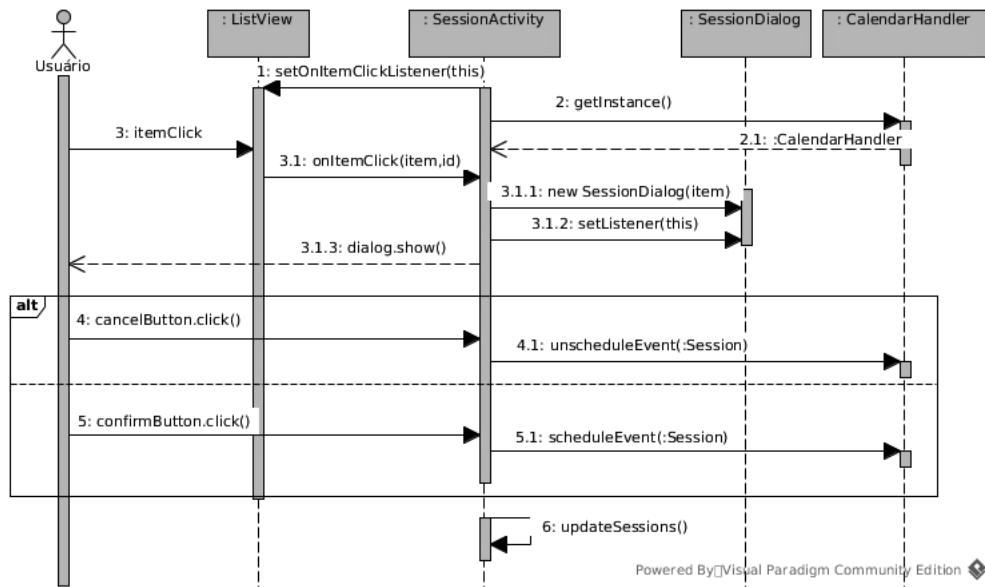


Figura 11: Diagrama de Sequência para Gerenciar Sessões.

Fonte: O autor.

A figura 12 especifica a sequência de mensagens trocadas ao realizar o agendamento do serviço que verifica os eventos em segundo plano, ela representa parte do caso de uso de verificar eventos em segundo plano. O serviço é agendado assim que o usuário se identifica no aplicativo ou quando o aparelho é reiniciado, sendo necessário para funcionamento da função de verificação de alterações em sessões.

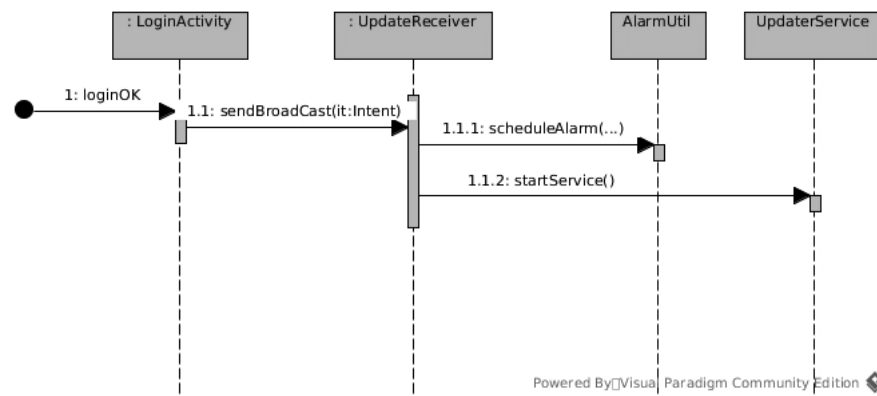


Figura 12: Diagrama de Sequência para agendar verificação de eventos.

Fonte: O autor.

A figura 13 especifica a sequência de mensagens trocadas entre as classes quando é marcado uma sessão na agenda, podendo a agenda estar livre para marcar eventos ou com eventos marcados no horário. Este diagrama representa parte das mensagens trocadas para execução do caso de uso de atualizar a agenda do Google no dispositivo.

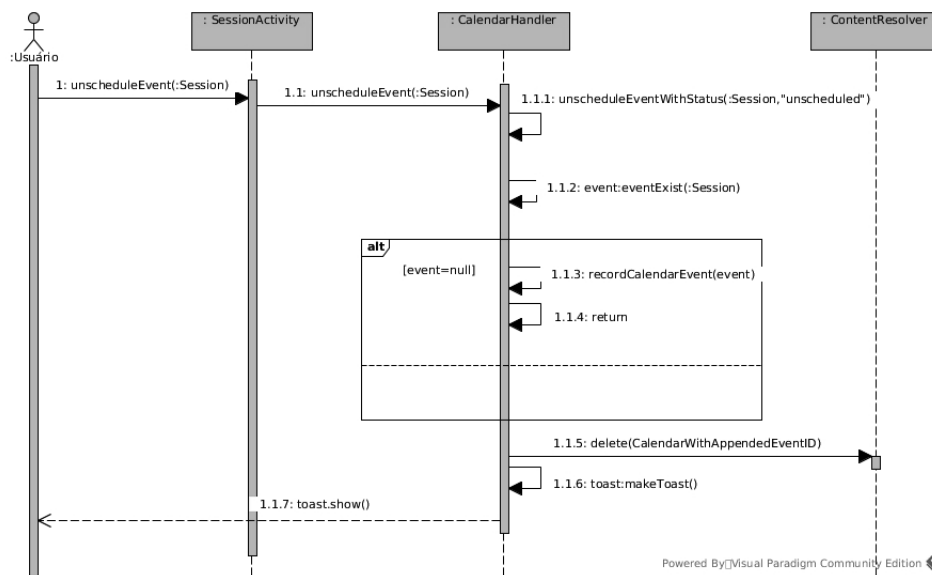


Figura 13: Diagrama de Sequência para atualizar agenda do Google no dispositivo - marcar.

Fonte: O autor.

A figura 14 especifica a sequência de mensagens trocadas entre as classes quando é desmarcado uma sessão na agenda, executando os procedimentos para remover a sessão da agenda do usuário. Este diagrama representa parte das mensagens trocadas para execução do caso de uso de atualizar a agenda do Google no dispositivo.

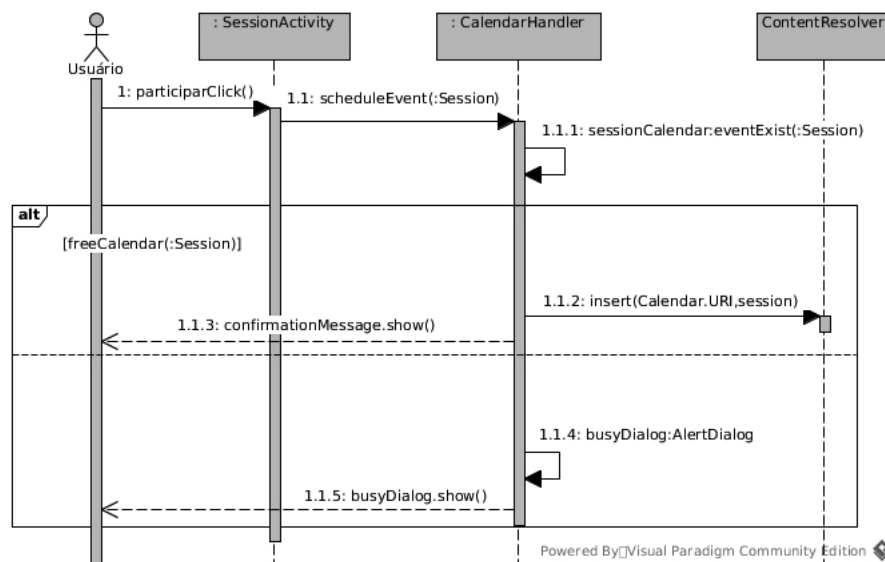


Figura 14: Diagrama de Sequência para atualizar agenda do Google no dispositivo - desmarcar.

Fonte: O autor.

A figura 15 especifica as mensagens trocadas entre as classes ao ser realizado a sincronização de eventos em segundo plano, este diagrama continua na figura 16, conforme o caso de

uso correspondente. As requisições à *API* são divididas em duas partes, uma para listar todos os eventos que o usuário está seguindo e outra para listar as sessões de cada evento seguido.

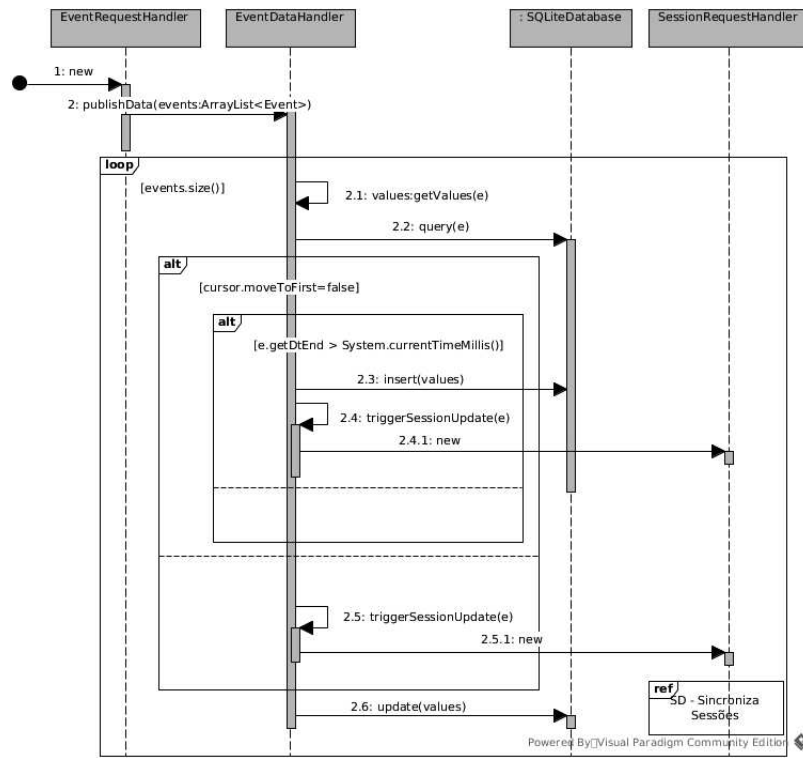


Figura 15: Diagrama de Sequência para processar eventos sincronizados.

Fonte: O autor.

A figura 16 especifica as mensagens trocadas durante a sincronização das sessões dos eventos seguidos pelo usuário, a atualização das sessões depende se houveram alterações nos dados da sessão. Este diagrama representa parte das mensagens trocadas para execução do caso de uso de verificar eventos em segundo plano.

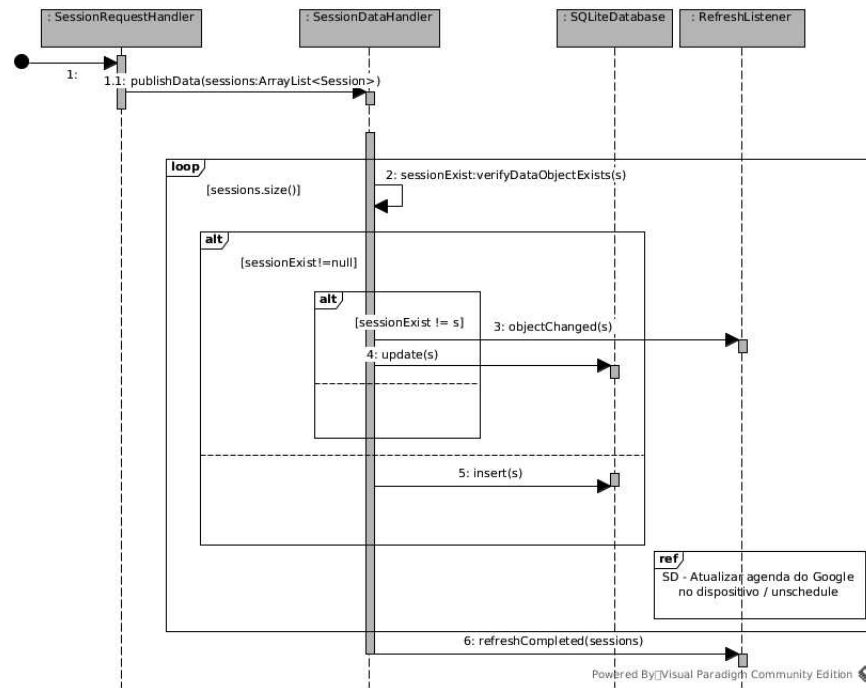


Figura 16: Diagrama de Sequência para processar sessões sincronizadas.

Fonte: O autor.

A figura 17 especifica as mensagens trocadas entre as classes que realizam a comunicação com a aplicação de gerenciamento de eventos utilizada, todas as ações são executadas em segundo plano. Os métodos utilizados da *API* estão especificados na documentação da aplicação de gerenciamento de eventos. Este diagrama representa parte das mensagens trocadas para execução do caso de uso de verificar novos eventos e de verificar eventos em segundo plano.

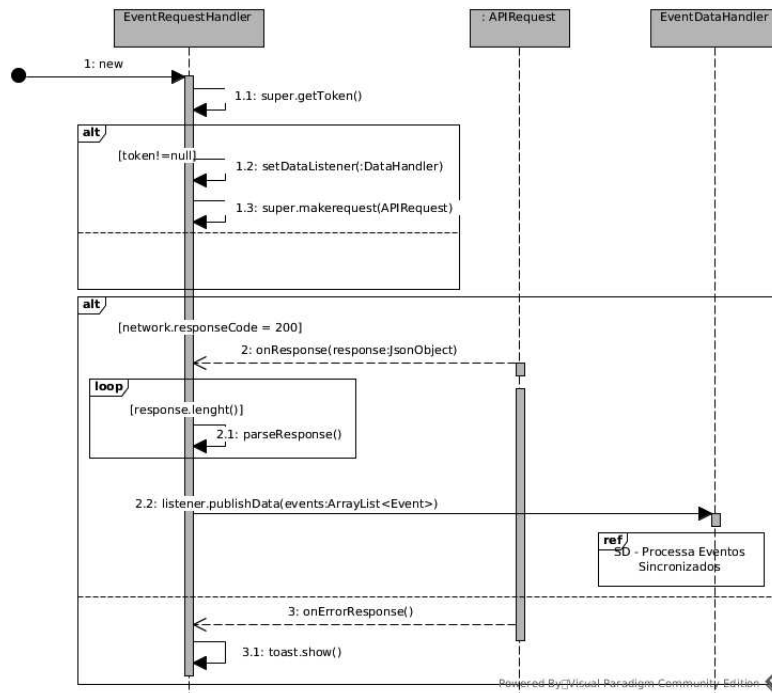


Figura 17: Diagrama de Sequência para sincronizar eventos.

Fonte: O autor.

A figura 18 especifica a sequência de mensagens trocadas entre as classes para processar as sessões de cada evento sincronizado, para cada evento seguido é disparado este fluxo de mensagens. Ao ser sincronizado o evento é enviado uma requisição à API para obter a lista de sessões. Este diagrama representa parte das mensagens trocadas para execução do caso de uso de verificar novos eventos e de verificar eventos em segundo plano.

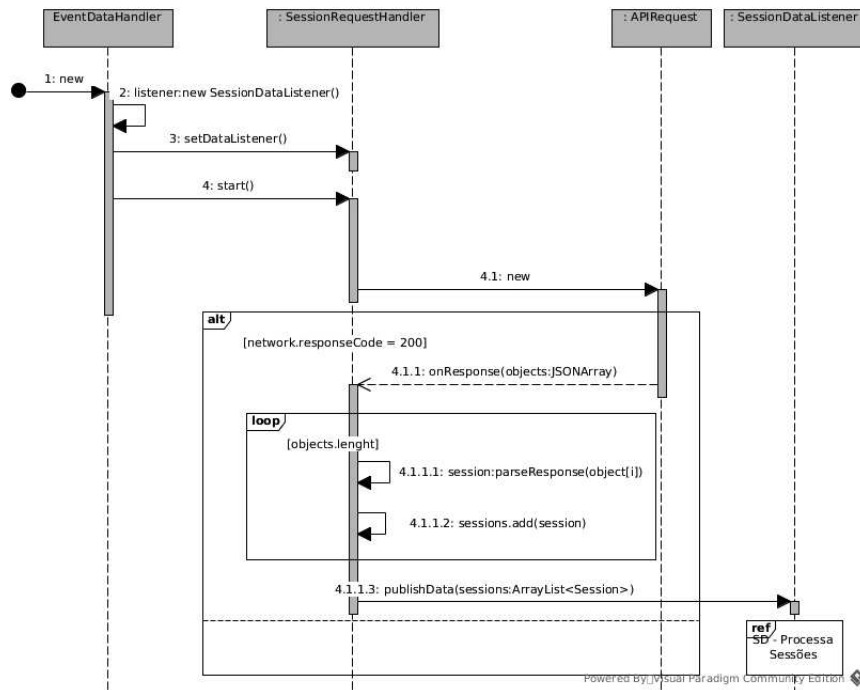


Figura 18: Diagrama de Sequência para sincronizar sessões.

Fonte: O autor.

A figura 17 especifica a sequência de mensagens trocadas para verificar eventos em segundo plano, ao ser identificada alteração no cronograma do evento é disparada a sincronização de eventos conforme figura 19. Este diagrama representa a parte inicial das mensagens trocadas para execução do caso de uso de verificar eventos em segundo plano.

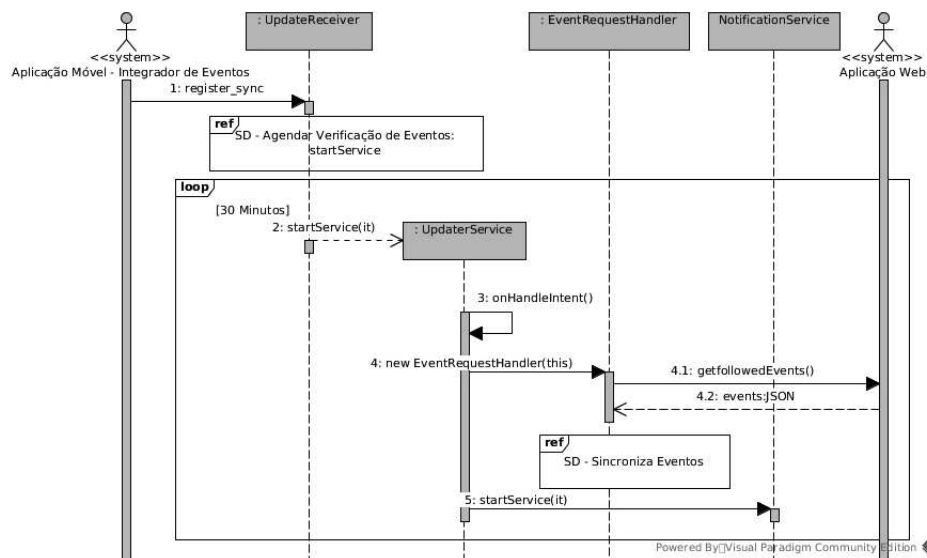


Figura 19: Diagrama de Sequência para verificar eventos em segundo plano.

Fonte: O autor.

A figura 20 especifica a sequência de mensagens trocadas entre as classes ao usuário solicitar a verificação de alteração manual nos eventos. Conforme o caso de uso com o mesmo nome.

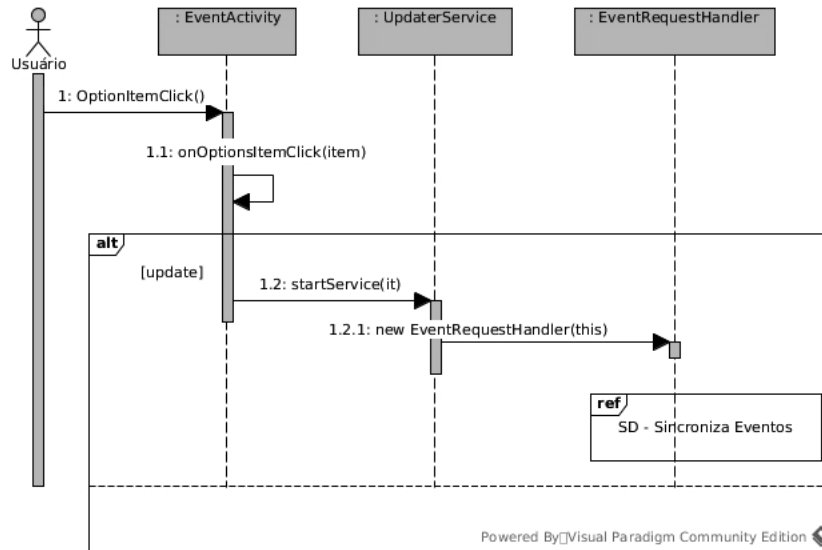


Figura 20: Diagrama de Sequência para verificar novos eventos.

Fonte: O autor.

A figura 21 especifica a sequência de mensagens trocadas entre as classes quando o usuário abre o aplicativo pela primeira vez ou quando a *token* precisa ser renovado, solicitando a identificação cadastrada na aplicação de gerenciamento de eventos. Este diagrama representa a execução o caso de uso de informar credenciais de acesso.

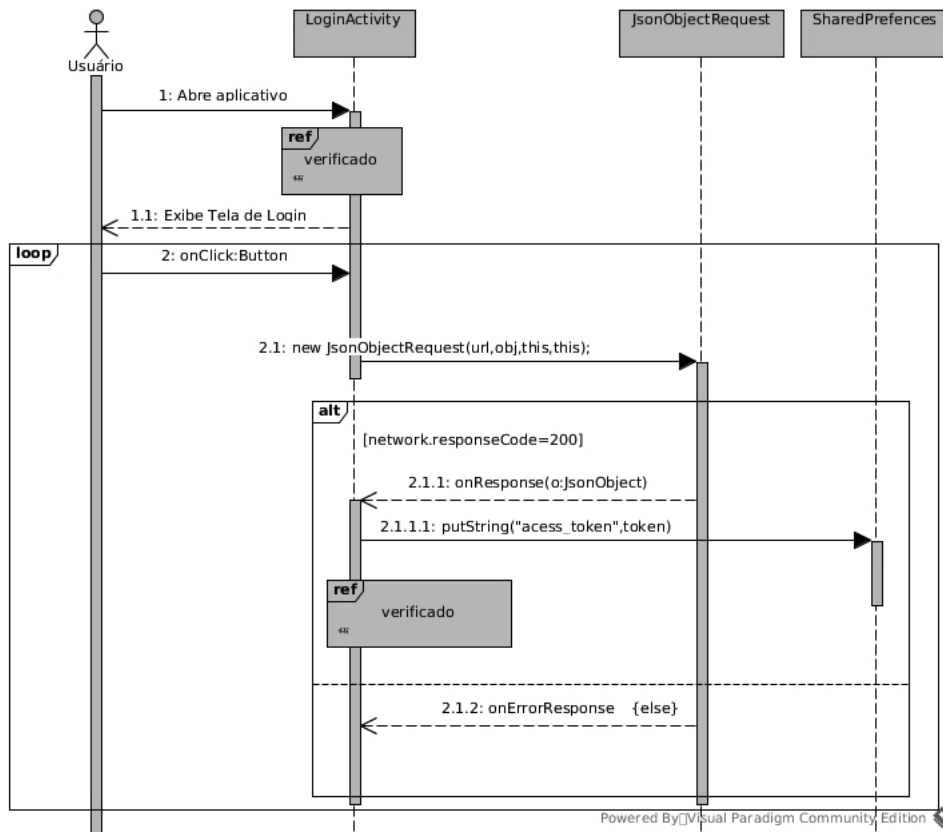


Figura 21: Diagrama de Sequência para informar credenciais de acesso.

Fonte: O autor.

A figura 22 representa a sequência de mensagens trocadas para verificar o *token* de acesso, este *token* é necessário para obter os eventos seguidos do usuário através da API disponibilizada pela aplicação de gerenciamento de eventos. Este diagrama representa parte das mensagens trocadas para execução do caso de uso de informar as credenciais de acesso, que foi dividido para facilitar a visualização do fluxo de mensagens.

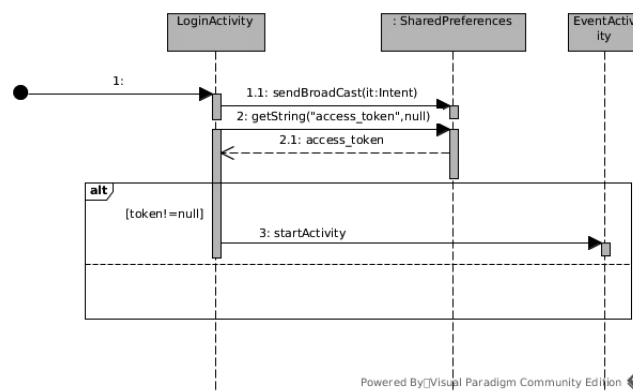


Figura 22: Diagrama de Sequência para verificar presença de *token*.

Fonte: O autor.

6 RESULTADOS

Foi desenvolvido um aplicativo móvel para a plataforma *Android* que sincroniza os horários dos eventos seguidos por um usuário na aplicação de gerenciamento de eventos escolhida. O aplicativo notifica e desmarca da agenda do *Google* do usuário as sessões quando ocorrerem mudanças nos horários, para que o mesmo remarque-as de acordo com sua disponibilidade. A inserção de uma sessão na agenda só é possível se o usuário não possuir nem outro compromisso agendado para o mesmo horário.

As telas a seguir demonstram o *software* desenvolvido e suas funcionalidades de sincronização, verificação de conflitos e inserção de sessão na agenda do *Google* do dispositivo. Foram inseridos dois eventos para fosse possível realizar testes e demonstrar as funcionalidades implementadas.

Ao aplicativo ser instalado é necessário informar as credenciais de acesso cadastradas na aplicação de gerenciamento de eventos, a figura 23 representa a tela de *login* desenvolvida. Solicitando ao usuário que seja digitado o *e-mail* e a senha do usuário.

Após informar as credenciais de acesso é exibido os eventos que o usuário marcou que deseja acompanhar na aplicação de gerenciamento de eventos. São exibidos, conforme a figura 24, os títulos dos eventos, uma breve descrição e o dia de início de cada evento.

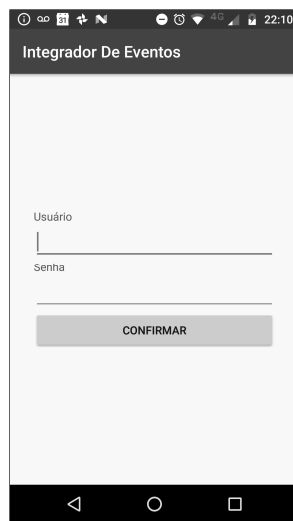


Figura 23: Tela de *Login*.

Fonte: O autor

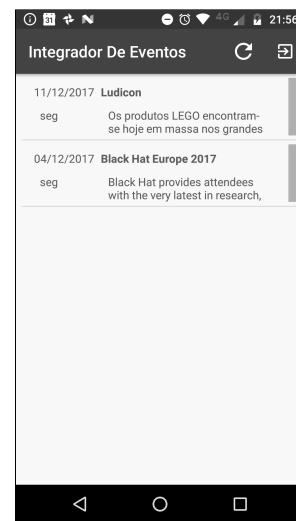


Figura 24: Tela de Visualização de Eventos Seguidos.

Fonte: O autor.

Ao pressionar sobre um dos eventos disponíveis na tela do aplicativo, é exibido as sessões que o evento possui, conforme figura 25. Esta tela possui uma breve descrição das informações mais importantes da sessão, possibilitando ao usuário visualizar o título, descrição breve, duração e data de início e *status* na agenda de cada sessão disponível no evento.

Para adicionar a sessão à agenda é necessário pressionar sobre uma das disponíveis na tela anterior, figura 25, abrindo uma tela de detalhes, figura 26. Nesta tela, além dos detalhes, é exibido dois botões, um para marcar que o usuário deseja participar da sessão, adicionando à agenda do usuário e ativando as notificações de alterações para a sessão marcada e outro para marcar que não deseja participar, desativando as notificações para esta notificação.

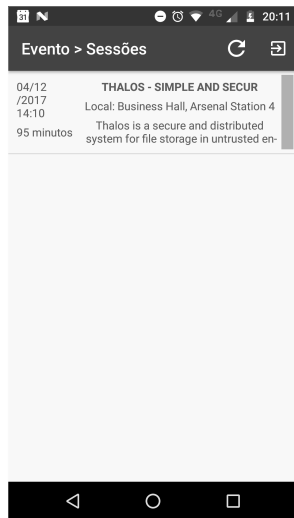


Figura 25: Sessões de Evento Sincronizado.

Fonte: O autor.



Figura 26: Detalhes de Sessão Adicionada.

Fonte: O autor.

Ao ser marcado que deseja ou não participar de uma sessão é realizado a alteração do *status* da mesma, esta mudança é informada pelo usuário por meio de uma notificação flutuante e de uma barra lateral que altera de cor conforme o *status* da sessão. A figura 27 representa uma sessão que o usuário deseja participar, sendo indicado pela cor verde. A figura 28 representa uma sessão de que o usuário não deseja participar, sendo indicada pela cor vermelha. Quando uma sessão recém foi adicionada ao evento ou que ainda não foi marcada com nenhum *status* pelo usuário as sessões ficam com a cor laranja, conforme figura 25.

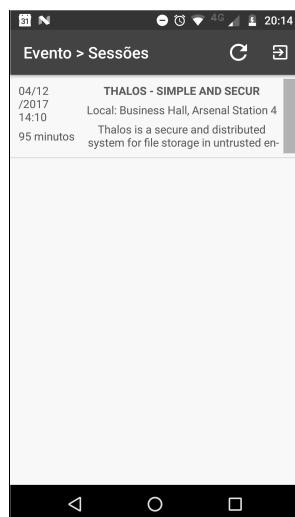


Figura 27: Estado sessão que usuário pretende participar.

Fonte: O autor.

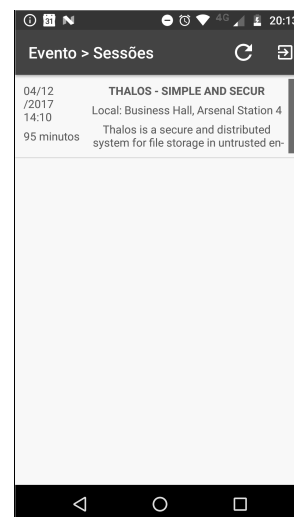


Figura 28: Estado de sessão que usuário não pretende participar.

Fonte: O autor.

Ao haver alguma alteração na programação do evento é gerado um notificação para o usuário de que a sessão que ele desejava participar trocou de horário e que é necessário reagendar ela. Para isto a notificação é mostrada ao usuário e ao clicar sobre a notificação o aplicativo é aberto no evento que sofreu alteração do seu cronograma, exibindo as sessões que foram alteradas.

Ao ser alterado alguma das sessões dos eventos seguidos é notificado o usuário os eventos que sofreram alterações, levando o usuário a abrir o aplicativo e remarcar as sessões que sofreram alterações ou verificar se deseja ou não participar de novas sessões adicionadas. A figura 29 representa a notificação gerada pelo aplicativo, esta notificação é exibida a cada 30 minutos caso o usuário abra o aplicativo e altere o *status* da sessão. Ao abrir o aplicativo as sessões que sofreram alterações serão exibidas com a cor laranja, conforme a figura 30.

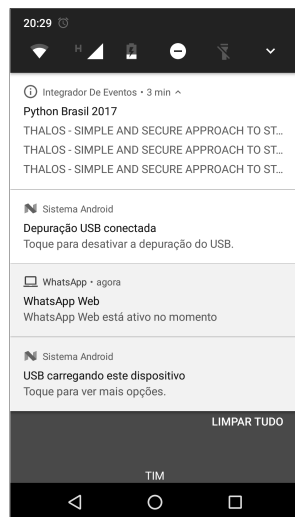


Figura 29: Notificação de Alteração de Sessão.

Fonte: O autor.

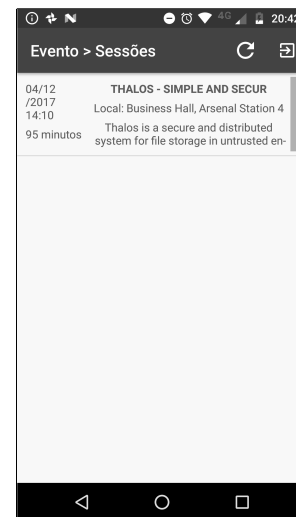


Figura 30: Estado de sessão após alteração.

Fonte: O autor.

A tela a seguir informa ao usuário que já existem sessões programadas para durante o horário que ele quer inserir a sessão em sua agenda, permitindo ao usuário realizar a inserção somente quando este conflito for solucionado.

Ao realizar o agendamento de uma sessão o aplicativo verifica na agenda do usuário se o mesmo possui eventos que sobrepõem a sessão a ser agendada, caso houver é exibido uma notificação alertando ao usuário de que existem outros eventos marcados no mesmo horário da sessão. A figura 31 representa esta notificação, enquanto o usuário não resolver este conflito de horários em sua agenda ele não poderá inserir a sessão através do aplicativo.

Quando um sessão é marcada que deve ser acompanhada, após a verificação de confli-

tos e constatação de que não existe nenhum evento que sobreponha o seu horário, ela é inserida à agenda do *Google* no dispositivo. A figura 32 representa a sessão que foi inserida na agenda do *Google* do usuário, são inseridos todos os dados da sessão para caso o usuário deseje verificar somente na agenda os eventos que estão marcados.



Figura 31: Tela informando conflitos de horários ao inserir em agenda sessão sobrepondo outra.

Fonte: O autor.

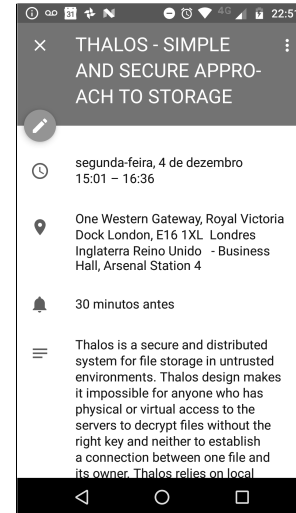


Figura 32: Tela de exibição de sessão inserida na agenda do usuário.

Fonte: O autor.

7 CONSIDERAÇÕES FINAIS

Ao desenvolver o *software*, percebeu-se a facilidade de identificar conflitos em horários de eventos que possuem diversas sessões simultaneamente e identificar alterações nos horários ao utilizar o aplicativo. Apesar da solução de gerenciamento de eventos encontrada para ser utilizada, verificou-se a necessidade de implementar em trabalhos futuros formas de buscar a programação dos eventos sem utilizar uma solução específica.

O trabalho contribuiu para o aprendizado quanto à forma de construir um *software*, desde as etapas de levantamento de requisitos e modelagem até a implementação. Propiciou, assim, uma consolidação dos conceitos vistos durante as disciplinas do curso, ampliando o conhecimento da linguagem de modelagem *UML* e de recursos que a mesma pode oferecer para facilitar a descrição e a visualização do *software* a ser construído.

Ao implantar a aplicação para gerenciamento de eventos foi necessário a modificação do *software*. Para isso, era preciso conhecer a forma de comunicação entre os componentes do *framework Flask*. Esta fase do trabalho contribuiu para o aprendizado deste *framework* e de seu funcionamento, permitindo, no futuro, escrever aplicações que o utilizem.

O desenvolvimento do aplicativo para a plataforma *Android* proporcionou uma visão mais aprofundada da plataforma utilizada e de seus componentes, qualificando para atuação de forma profissional. Ao desenvolver o *software* foi necessário estudar formas de comunicação com outras aplicações na Internet e como tratar erros nas mesmas, o que possibilita a capacidade de criar aplicações mais dinâmicas.

O hábito de consultar a documentação de *APIs* mostrou o quão importante é documentar os métodos disponíveis e a sua forma de utilização. Colaborando para que os projetos futuros sejam cada vez melhor documentados, conforme a experiência adquirida.

REFERÊNCIAS

- ADVICE, S. **Página com alternativas de Softwares de Gerenciamento de Eventos**. 2017. Disponível em: <<https://www.softwareadvice.com/event-management/>>. Acesso em: 16 nov. 2017.
- BOOCH, G. **UML guia do usuário**. [S.l.]: ElsevierCampus, 2012. ISBN 9788535217841.
- EVENTBASE. **EventBase website**. 2017. Disponível em: <<https://www.eventbase.com/>>. Acesso em: 08 jun. 2017.
- EVENTBRIDE. **Eventbride website**. 2017. Disponível em: <<https://www.eventbrite.com.br/>>. Acesso em: 08 jun. 2017.
- FOSSASIA. **Repositório de Software de Gerenciamento de Eventos: Open Event Orga Server - Fossasia**. 2017. Disponível em: <<https://github.com/fossasia/open-event-orga-server>>. Acesso em: 09 jun. 2017.
- GITHUB. **Repositório de códigos-fonte**. 2017. Disponível em: <<http://github.com>>. Acesso em: 17 nov. 2017.
- GLOBALWEBINDEX. **Pesquisa sobre o uso de Smartphones na Internet segundo a GlobalWebIndex**. 2017. Disponível em: <<https://www.globalwebindex.net/blog/94-of-16-24s-own-a-smartphone>>. Acesso em: 05 abr. 2017.
- GOOGLE. **Página de visualização de estatísticas de smartphones que utilizam Android**. 2017. Disponível em: <<https://developer.android.com/about/dashboards/index.html?hl=pt-br>>. Acesso em: 16 nov. 2017.
- HELLMANN, D. **virtualenvwrapper 4.7.3.dev7**. 2017. Disponível em: <<https://virtualenvwrapper.readthedocs.io/en/latest/>>. Acesso em: 17 jun. 2017.
- HEROKU. **Descrição da plataforma Heroku**. 2017. Disponível em: <<https://www.heroku.com/about>>. Acesso em: 15 nov. 2017.
- NEUSTAEDTER, C.; BRUSH, A. J. B.; GREENBERG, S. The calendar is crucial: Coordination and awareness through the family calendar. **ACM Trans. Comput.-Hum. Interact.**, ACM, New York, NY, USA, v. 16, n. 1, p. 6:1–6:48, abr. 2009. ISSN 1073-0516. Disponível em: <<http://doi.acm.org/10.1145/1502800.1502806>>.
- POCOO. **Documentação oficial do framework Flask**. 2017. Disponível em: <<http://flask.pocoo.org/>>. Acesso em: 15 nov. 2017.
- SANCHEZ, S. **A Importância do Planejamento Pessoal**. 2015. Disponível em: <<http://vivabemavida.com.br/coaching/life-coaching/a-importancia-do-planejamento-pessoal/>>. Acesso em: 31 mai. 2017.

SMITH, B. **A Quick Guide to GPLv3**. 2014. Disponível em: <<https://www.gnu.org/licenses/quick-guide-gplv3.html>>. Acesso em: 17 jun. 2017.

SOFTWARE, O. **Estatísticas de uso de celular no Brasil**. 2016. Disponível em: <<https://www.opus-software.com.br/estatisticas-uso-celular-brasil/>>. Acesso em: 31 mai. 2017.