| Faculdades Integradas de Taquara - Faccat<br>Av. Oscar Martins Rangel, 4.500<br>Taquara, RS, CEP 95600-000 | ***Curso de Sistemas para Internet*** |
| --- | --- |

# SERVIÇO DE TICKETS ELETRÔNICOS PARA EVENTOS

Gabriel da Silva Silveira
Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
gsilveira.rs@gmail.com

Leonardo Ribeiro Machado
Professor Orientador
Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
leonardomachado@faccat.br

**Resumo**

Este artigo tem por objetivo apresentar como foi o processo de desenvolvimento da ferramenta e-Tickets através de uma abordagem técnica mostrando os pontos mais importantes sobre o desenvolvimento e mesmo sobre modelo de negócio. O serviço implementa um conceito relativamente novo em vendas online chamado de market place por meio de compra de ingressos online, cenário onde o vendedor e o comprador são ligados por intermédio do serviço.

**Palavras-chave**: eventos, *web service*, aplicativo híbrido, *ticket* eletrônico.

## ELETRONIC TICKETING SERVICE FOR EVENTS

***Abstract***

*This article aims to present the development process of e-Tickets tool, through a technical approach showing the most important points on the development and even on the business model. The service implements a relatively new concept in online sales called market place through purchasing online tickets, scenario where the seller and the buyer are connected through the service.*

***Keywords:*** *events, web service, hybrid application, electronic ticket.*

## 1 Introdução

A área de eventos tem crescido muito nos últimos anos. Certamente, em breve o que vai diferenciar um evento de outro é a qualidade de sua organização. Entre tantos itens envolvidos em um evento, um dos que pode causar muitos problemas é o ingresso. Dúvidas sobre como e onde comprá-lo, se é possível efetuar a comprar utilizando cartão de crédito ou apenas boleto bancário**,** como é feita entrega do ticket, em quando tempo será entregue, configuram uma série de perguntas feitas pelo consumidor, e que muitas vezes não são respondidas pelo fornecedor.

Com o avanço do uso de compra e venda online (comércio eletrônico), a área de eventos também foi beneficiada. Atualmente, existem muitos sites ou aplicativos direcionados para a compra de ingressos, mas ainda muitas dúvidas são geradas e não respondidas. Tudo isto pode causar uma série de problemas, que muitas vezes não são solucionados em tempo hábil, gerando transtornos tanto para o fornecedor quanto para o cliente.

A partir disso, conclui-se que um aplicativo ágil e eficiente, que possua um calendário com os vários eventos classificados corretamente em sua tipologia, área de interesse e categoria, para facilitar uma rápida visualização e escolha por parte do cliente, poderia solucionar grande parte dos problemas que foram apresentados.

Este artigo documenta a construção deste aplicativo, o qual tem a proposta de suprir esta lacuna. O artigo demonstra a problemática envolvida no tema pesquisado e detalha os passos que foram percorridos na construção da ferramenta, desde a definição da metodologia a ser utilizada, passando pela análise, projeto e construção da mesma. Ao fim, são demonstrados os resultados obtidos com o seu desenvolvimento.

A seção 2 abaixo traz um referencial teórico relacionado aos assuntos pesquisados e abordados na solução. A seção 3 demonstra ferramentas relacionadas à que foi desenvolvida ainda mostra, passo a passo, a construção da ferramenta através da metodologia de desenvolvimento selecionada. A seção 4 demonstra os resultados obtidos, e a seção 5 faz uma conclusão acerca da pesquisa que foi realizada.

## 2 REFERENCIAL TEÓRICO

### 2.1 Eventos

Segundo Houaiss (2010, p. 337), um evento pode ser definido como "qualquer acontecimento (festa, espetáculo, comemoração, solenidade etc.) organizado por especialistas, com

objetivos institucionais, comunitários ou promocionais". Sobre essa ideia, a definição de evento torna-se ampla e complexa.

O evento é um instrumento institucional e promocional, utilizado na comunicação dirigida, com a finalidade de criar conceitos e estabelecer a imagem de organizações, produtos, serviços, ideias e pessoas, por meio de um acontecimento previamente planejado, a ocorrer em um único espaço de tempo com a aproximação entre os participantes, quer seja física, quer seja por meio de recursos da tecnologia (MEIRELLES, 1999).

Cesca (1997) afirma que um evento desperta a curiosidade do público e promove os organizadores.

De acordo com Meirelles (1999), uma reunião caracteriza-se como o embrião de todos os tipos de eventos, e trata-se do encontro de duas ou mais pessoas, a fim de discutir, debater e solucionar questões sobre determinado tema relacionado com suas áreas de atividade. Pode-se ainda dizer que toda reunião é um evento, pois irá necessitar de planejamento e de uma infraestrutura para alcançar as metas propostas.

Meirelles (1999) ainda nos relata a tipologia de eventos tendo como base a reunião:

- Reunião dialogal: baseada na informação, no questionamento e na discussão - palestra, conferência, seminário, simpósio, convenção, entrevista, entre outros;
- Reunião coloquial: baseada no entretenimento, no lazer, na aproximação entre as pessoas e na confraternização - coquetel, café da manhã, almoço, jantar, *brunch*, *happy hour,* shows, entre outros;
- Competitivas: concursos, torneios, entre outros;
- Expositivas e Demonstrativas: feira, salão, mostra, exposição, desfile, lançamento de produtos, inauguração, entre outros.

Como a abrangência de eventos é muito vasta, devemos ter em mente outras maneiras de classificação, como área de interesse e categoria, por exemplo. Uma possível divisão seria categorizar-se os eventos como sendo eventos especiais, de participação, permanentes, esporádicos, únicos, de oportunidade, de massa, de nicho, promocionais de marca, promocionais de produtos e serviços, locais, regionalizados ou globais (MEIRELLES, 1999).

## 2.2 Comércio Eletrônico

A expressão e-Commerce pode ser traduzida como “compras eletrônicas” ou ainda "comércio eletrônico". No início (cerca de 1970), foi algo muito simples, com transferências de

valores entre pessoas e empresas, mas com o crescente uso da internet o comércio eletrônico foi ganhando força, confiabilidade e disponibilidade, estando ao alcance de todos. A princípio o comércio eletrônico foi utilizado para viabilizar a compra e venda de livros e CDs através da Internet. Com o passar do tempo, no entanto, o quadro mudou totalmente, pois atualmente um grande diferencial do comércio eletrônico é a possibilidade de atender a vários tipos de modelos de negócios, tanto a grandes empresas quanto a um pequeno empreendedor de algum nicho mais específico (MENDES, p. 11).

Em função de a Internet ser considerada um meio de comercialização poderoso, milhares de usuários são expostos à comunicação e às estratégias de marketing das empresas online. O número de pessoas que estarão conectadas nesse meio, num espaço razoável de tempo, pode atingir um patamar elevado de consumidores, quando comparadas a divulgação e estratégias de seu negócio por outros meios de comunicação (COELHO, 2011).

O ato de comprar é comportamental. A compra online é um comportamento novo e, ainda, está consolidando-se no dia-a-dia da maior parte dos usuários da web. Na compra realizada na internet, o consumidor percebe a comodidade e segurança nas transações de dados pessoais no ambiente online, beneficios que agregam valor ao negócio e fazem o usuário retornar, é o que afirmam Silva e Jacob (2011).

De acordo com Felipini (2012), o número de pessoas que estão substituindo as compras tradicionais pela alternativa móvel não para de crescer, e esta é uma tendência do mercado para agora e para o futuro. Felipini (2012) ainda conclui que aqueles que não aderirem ficarão estacionados no tempo e perderão espaço no mercado, que está em constante transformação. Neste cenário é preciso pensar menos em emprego e mais em trabalho.

Nos últimos anos, a quantidade de usuários de comércio eletrônico passou de um milhão em 2001 para 61,5 milhões até 2014, sendo que hoje no Brasil há mais de 100 milhões de internautas. Isso deve-se ao surgimento das redes 3G e 4G, smartphones e outros dispositivos portáteis de tecnologia avançada, e ainda cada vez mais popular é o uso destes dispositivos para compras na internet. Em 2014 foram concretizados mais de 103,4 milhões de pedidos através da Internet. Pelo menor custo, pela comodidade e também pela inclusão digital das classes C, D e E, de acordo com a E-bit, empresa especializada em comércio eletônico, o setor movimentou R$ 35,8 bilhões em 2014, com um crescimento de 24% comparado a 2013, quando teve seu faturamento em torno dos R$ 28,8 bilhões. Para 2015, a empresa prevê um crescimento de 20%, para que o faturamento atinga um valor estimado de R$ 43 bilhões (SEBRAE, 2015).

Outro ponto positivo para se empreender na internet através de comércio eletrônico é que este ramo é relativamente novo. Por este motivo, ainda há uma procura enorme por conhecimento da receita de sucesso na internet, e isto acaba nivelando as coisas, tornando a diferença entre quem acaba de entrar neste ramo com quem já trabalha nele muito pequena. Isto, aliado à quantidade de nichos de mercados inexplorados e à facilidade de se conseguir montar uma loja virtual, torna-se uma grande oportunidade para os novos empreendedores (FELIPINI, 2012).

## 2.3 Compra de ingressos pela internet

Segundo Churchill (2003), o transcurso de comprar é influenciado por motivos situacionais, sociais e de marketing, e o resultado é que para cada consumidor há uma necessidade especial para ser suprida. A facilidade que existe na distribuição de bens e serviços na internet, a não necessidade de estar geograficamente presente em uma loja física, e o fato de os valores serem mais acessíveis, são itens que encantam os usuários (COELHO, 2015). Um site de vendas deve ser rápido e intuitivo para o usuário. Este grau de facilidade é um dos itens que vai determinar a sua boa aceitação ou não, perante o usuário. No Brasil, em 1995, uma organização estabelecida com a finalidade de vender ingressos de cinema, shows e teatro, foi criada. Trata-se da "ingresso.com.br", a pioneira neste segmento no Brasil. Ela segue o tipo de serviço em que o usuário compra o ingresso e recebe em casa ou vai buscar nos pontos de venda/troca (FERREIRA, CHAUVEL e SILVEIRA, 2006).

## 2.4 Metodologia e Ferramentas

### 2.4.1 Scrum

Scrum é um framework estrutural usado para gerenciar produtos e projetos de alta complexidade desde o inicio da década de 90. Scrum não é um processo ou técnica de construção de produtos, pelo contrário, é uma ferramenta que ajuda a empregar processos ou técnicas. Melhoria continua de um projeto, frequente resolução de problemas e aumento da produtividade são vantagens que se obtem ao usar Scrum (GUERRATO, 2013).

Schwaber e Sutherland (2013) mostram que o Scrum é fundamentado nas teorias empíricas de controle de processo, onde se afirma que o conhecimento vem da experiência e de tomadas de decisões com base em dados históricos. Scrum emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos.

Times Scrum são auto-organizáveis e multifuncionais. Cabe a esses times escolherem a maneira de completar seu trabalho. No time Scrum todos têm os mesmos valores, assim como todas

as opiniões são válidas. o time é composto pelo Product Owner, Scrum Master e o Time de desenvolvimento, cujo tamanho ideal é de sete pessoas, mais ou menos duas pessoas. Times com menos de cinco pessoas têm menor produtividade e podem encontrar dificuldades durante as sprints (cada iteração ou unidade básica de desenvolvimento usando-se scrum). Já um time com mais de nove pessoas acaba tornando-se muito complexo para controlar, mas há casos bem-sucedidos de times que extrapolaram este limite (SCHWABER, p. 8, 2009).

### 2.4.2 PHP

A linguagem PHP foi concebida durante o outono de 1994 por Rasmus Lerdorf. As primeiras versões não foram disponibilizadas, tendo sido utilizadas em sua homepage apenas para que ele pudesse ter informações sobre as visitas que estavam sendo feitas. A primeira versão utilizada por outras pessoas foi disponibilizada em 1995, e ficou conhecida como "*Personal Home Page Tools*" (ferramentas para página pessoal). Era composta por um sistema bastante simples que interpretava algumas macros e alguns utilitários que executavam por trás das homepages: um livro de visitas e um contador, entre outros (DARLAN, 2007).

Graças a ferramentas como o Facebook e Instagram, o PHP teve um grande aumento em sua comunidade de desenvolvedores, e deixou de ser uma simples linguagem para sites pequenos para tornar-se uma linguagem robusta com ampla utilização no mercado.

### 2.4.3 IONIC

Um aplicativo hibrido é aquele que é todo desenvolvido usando tecnologias de desenvolvimento web, como HTML, JavaScript e CSS, e, pós o desenvolvimento, esta aplicação é traduzida para a linguagem nativa do sistema operacional mobile escolhido (AMBROS, 2013). No caso do sistema operacional Android, esta linguagem é o Java. Isso só é possivel porque todo código do aplicativo é executado dentro do app nativo através de um *webview*, nome dado a um tipo simples, porém especial, de navegador que é executado assim que a execução do aplicativo híbrido é iniciada no dispositivo. Isto fica transparente ao usuário, de forma que ele não tem como reconhecer se o aplicativo é hibrido ou não de forma intuitiva. (RAVULAVARU, 2015).

O grande benefício da utilização do *webview* é que ele não se limita apenas a renderizar o aplicativo: ele também dá suporte para o acesso a recursos nativos do dispositivo, como câmera, microfone, acelerômetro, entre outros Isto é possível graças a uma interface JavaScript que faz com que a *webview* seja apta para executar tais códigos nativos dos dispositivos (CHEDE, 2013).

Há duas plataformas principais utilizadas no desenvolvimento híbrido: a plataforma Cordova e a plataforma Phonegap. Através destas plataformas são criados os aplicativos nativos com uma *webview*, que por sua vez roda o HTML, o CSS e o JavaScript criados pelos desenvolvedores. As plataformas também baseiam-se no conceito de plugins, para acesso aos diversos recursos dos dispositivos (VASCONELLLOS, 2015).

O IONIC é um SDK (*software development kit)* de componentes visuais focados em aplicativos, e abstrai toda a complexidade das funções e plugins do Cordova. Além disso, incorpora o framework JavaScript AngularJs, que é mantido pelo Google e é muito difundido pelo mundo. Com uma abordagem onde o foco principal é criar uma SPA (*Single Page Application*), o AngularJs facilita muito o desenvolvimento de aplicativos por usar o conceito de *Two-way-databinding*. Também torna-se uma opção mais barata e mais rápida do que a opção de desenvolver aplicativos nativos. Outro grande diferencial do IONIC é a sua performance, quando comparado a outras frameworks da mesma categoria, o que torna o IONIC uma bela alternativa ao desenvolvimento nativo (Vasconellos, 2015).

### 2.4.4 Web Services

Segundo a W3C (2004), um web service é um sistema desenvolvido para suportar a comunicação entre máquinas diferentes de distintas plataformas, geralmente usando protocolos como HTTPS, HTTP, SMTP, FTP, RMI/IIOP ou protocolos de mensagem proprietários em conjunto com mais alguns padrões web impostos pela mesma.

Uma interface bastante utilizada é a REST, baseada nos protocolos HTTP e HTTPS. Segundo Fielding (2000), para um web service utilizar a interface REST, é necessário que sejam seguidas as seguintes restrições de funcionamento:

- Cliente-Servidor - onde um servidor com serviços à disposição escute requisições para estes serviços a partir de algum cliente;
- Sem estado (Stateless) - onde a comunicação Cliente-Servidor deve ser feita sem criar qualquer tipo de estado no servidor;
- Cache - forma para aumentar o desempenho da aplicação quando os dados de uma requisição que ser mantidos de forma temporária.

O protocolo HTTP é baseado em texto, e gira em torno do cliente e do servidor, os quais são responsáveis pela troca de mensagens para o funcionamento do protocolo. Por definição, o cliente sempre inicia a conversa para então o servidor responder. Por ser baseado em texto, as mensagens

são uma fração de bits de dados, porém o corpo da mensagem pode conter arquivos de mídia. A partir de um *header* e de um *body,* são feitas as mensagens do protocolo. No *header* são guardadas as informações de metadata, como por exemplo o Content-type e mais uma série de métodos HTTP (Grigorik, p. 222, 2013).

Através das URLs, identificamos os recursos únicos de uma aplicação REST para executar ações através dos verbos HTTP, pois cada requisição é atrelada a um verbo HTTP. São eles que dizem ao servidor o que fazer com o recurso identificado na URL, podendo a requisição ter informações adicionais que eventualmente são alocadas no *body*. Dentre os verbos *HTTP,* estão o *GET,* o *POST,* o *PUT* e o *DELETE* (Fischer, 2013).

## 3. Desenvolvimento da Ferramenta

A partir dos problemas discutidos no referencial teórico a respeito do cenário atual sobre o comércio eletrônico e, mais especificamente, sobre a compra de ingressos de forma on-line, e baseando-se nas tecnologias discutidas na seção 2.4, partiu-se para o desenvolvimento da ferramenta para compra on-line de ingressos.

A ferramenta desenvolvida foi chamada de e-Tickets, e é composta por alguns módulos, que possuem funções bem definidas, e que trabalham em conjunto para o funcionamento da mesma, produzindo ao final o resultado esperado de uma ferramenta que permita ao usuário efetuar a compra de ingressos para eventos de forma on-line, sempre levando em consideração os diferentes detalhes dos eventos envolvidos, como categorização e área de interesse, por exemplo.

Esta seção descreve a construção da ferramenta e-Tickets, detalhando como os aspectos de arquitetura, metodologia e tecnologia retratados na subseção 2.6 foram utilizados para esta construção.

### 3.1 Ferramenta e-Tickets

Com uma análise feita através de levantamento de requisitos, diagramas de casos de uso, diagrama de classes, diagrama de implantação e um plano de desenvolvimento de software, foi possível reunir o conhecimento necessário para as definições das tarefas. O projeto contou com um processo de desenvolvimento ágil e unificado utilizado através da metodologia Scrum.

Para o gerenciamento das sprints e suas tarefas foi usado o software Trello. Através da análise do que deveria ser desenvolvido, foi decidido dividir o desenvolvimento entre os segmentos *web service* e aplicativo. O *web service* é separado nos módulos estrutura, autenticação, recursos,

persistência, compras e tickets. O aplicativo está dividido nos módulos autenticação, rotas, eventos, compra e tickets. Para o desenvolvimento destes módulos foram impostas uma série de tarefas que estão divididas em 4 iterações, cada uma de 30 dias de duração, as quais levaram a um total de 120 dias de desenvolvimento.

O Quadro 1 mostra a separação dos módulos e das tarefas respectivas dentro das Sprints e logo abaixo a figura 1 apresenta uma visão geral de como a ferramenta funciona.

QUADRO 1 - Sprints e suas tarefas

| Sprint | Segmento | Módulo | Tarefa | Descrição |
|---|---|---|---|---|
| 1 | Web Service | Estrutura | Criar redirecionamentos. | Criar arquivo com regras de servidor para redirecionar as requisições ao núcleo do *web service*. |
| 1 | Web Service | Estrutura | Criar estrutura do *web service*. | Criar a estrutura da API usando o seguinte o design. Isolar as requisições separando os recursos e identificando as ações através dos verbos HTTP. |
| 1 | Web Service | Recursos | Criar acesso ao recursos através de classes e metódos para os mesmos. | Criar classes para a representação dos recursos e métodos para representar ações que cada recurso oferece. |
| 1 | Web Service | Persistência | Criar a lógica para persistência de dados dos recursos. | Criar estrutura de dados para serem persistidos no banco de dados. |
| 1 | Web Service | Autenticação | Autenticar Clientes e Usuários | Implementar a bliblioteca OAuth usando do tipo *User Credentials*. Criar credenciais de Cliente para acesso do Aplicativo. |
| 2 | Aplicativo | Estrutura | Criar estrutura do aplicativo. | Criar a estrutura do aplicativo através da CLI do IONIC *framework*. |
| 2 | Aplicativo | Rotas | Criar as rotas. | Criar rotas para as páginas do aplicativo. |
| 2 | Aplicativo | Autenticação | Criar Fluxo | Criar o fluxo de autenticação do tipo *User Credentials* para o acesso ao serviço conforme seu tipo de usuário (Usuário normal ou *Promoter*). |
| 2 | Aplicativo | Eventos | Criar Fluxo | Criar fluxo criação e divulgação do evento. |
| 3 | Aplicativo | Compras | Criar Fluxo | Implementar pedido de compra, e *checkout* de dados. |
| 3 | Web service | Compras | Implementar biblioteca Pagar.me | Implementar o compra através do da biblioteca do Pagar.meJs |
| 4 | Aplicativo | Tickets | Criar e-Tickets | Implementar ticket eletrônico usando criptografia assimétrica para proteger o código do ingresso. |
| 4 | Web service | Tickets | Criar e-Tickets | Implementar ticket eletrônico usando criptografia assimétrica para proteger o código do ingresso. |

Fonte: Autoria própria (2015)

A figura 1 abaixo demonstra, de forma genérica, a arquitetura e os componentes da ferramenta desenvolvida. Trata-se de um diagrama de implantação, construído com a utilização da linguagem UML (Unified Modelling Language), que tem por finalidade demonstrar como a ferramenta está organizada, e como ocorre a comunicação entre as diferentes partes da solução.

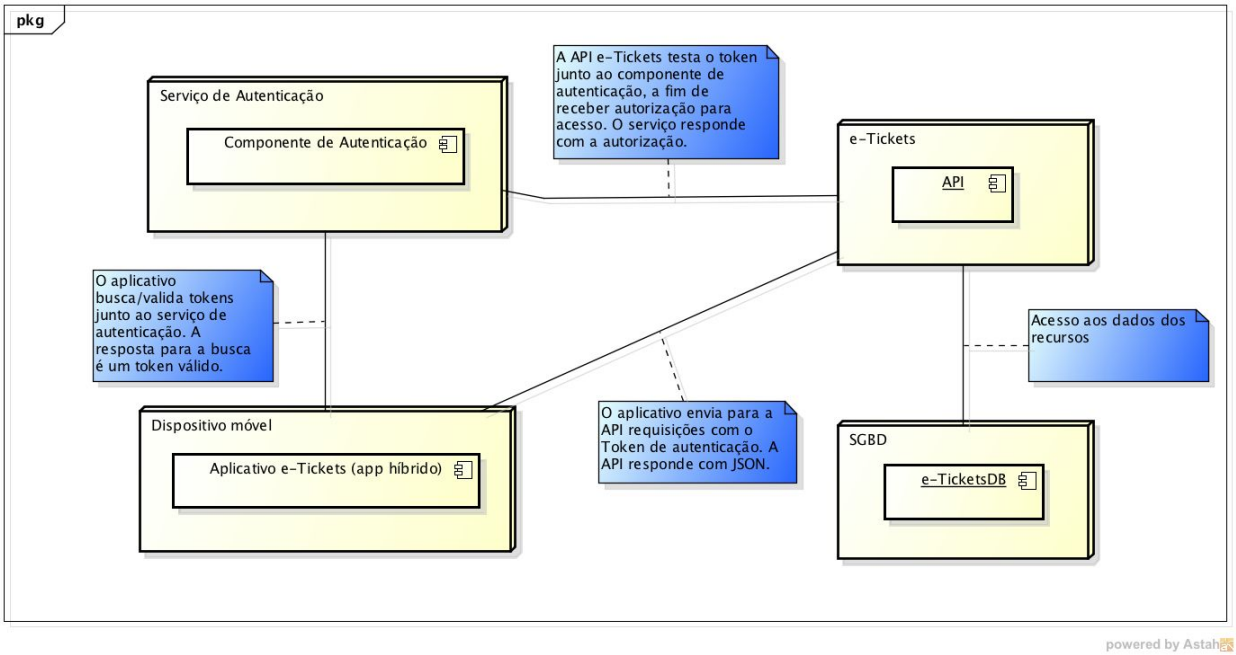


Figura 1 - Visão geral de funcionamento.
Fonte: Autoria própria (2015)

### 3.2 Web service

Esta seção descreve, de forma técnica, a maneira como foi implementado o web service utilizado na ferramenta. Este entendimento é importante para as seções que seguem.

O desenvolvimento da ferramenta foi feito através da linguagem de *script* para servidores PHP, utilizando o MySql como SGBD (sistema gerenciador de banco de dados). Antes de mais nada, foi preciso tomar cuidado quanto à estrutura da aplicação, para que a mesma desempenhe suas funções com facilidade e destreza. É necessário o uso de regras de servidor para tornar as URLs das requisições amigáveis e com isso redirecionar todas as requisições para o núcleo da aplicação no servidor. Este tipo de montagem é indicado pois, através dela, obtém-se maior facilidade na hora de executar ações sobre os recursos do serviço.

Este tipo de serviço é baseado principalmente no funcionamento de dois *web standards,* o HTTP e o URI, onde através das URLs podemos identificar recursos dinamicamente, e juntamente aos verbos HTTP aplicar ações a estes recursos, criando um design de serviço no estilo REST. Isto permitiu que fossem agregados benefícios ao sistema, como escalabilidade, manutenibilidade, comunicações sem estado, todos os recursos com identificadores, recursos com multi

representações, e ainda vinculação de recursos. A Figura 1, exibida abaixo, detalha a estrutura do *web service* desenvolvido.

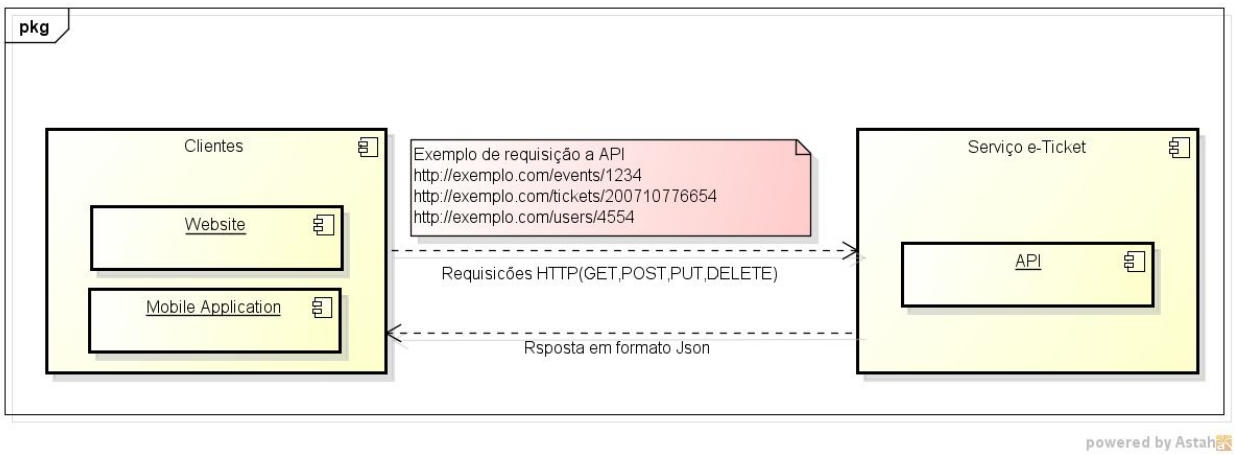


Figura 2 - Exemplo de funcionamento do Serviço.
Fonte: Autoria própria (2015)

Conforme a Figura 1 acima, podemos notar que o serviço funciona "escutando" as requisições de seus clientes, identificando os recursos envolvidos (pode ser mais de um recurso envolvido através da vinculação de recursos) e processando as informações recebidas para então responder ao cliente que fez a requisição. Para cada recurso temos até 4 ações (criar, pegar, atualizar e deletar) implementadas através de lógica de programação, conforme mostrado na Quadro 2 abaixo.

QUADRO 2 - Recursos

| Ação | Verbo HTTP | Recursos | URL |
|---|---|---|---|
| Criar | PUT | Eventos, Promoters, Users, Compras | http://www.etickets.eti/nome-do-recurso |
| Pegar | GET | Locais, Tickets, Seguidores, Promoters, Compras, Users, Locais | http://www.etickets.eti/nome-do-recurso ou http://www.etickets.eti/nome-do-recurso/id-do-recurso |
| Atualizar | POST | Eventos, Promoters, Users, Compras, Locais. | http://www.etickets.eti/nome-do-recurso/id-do-recurso |
| Deletar | DELETE | Eventos | http://www.etickets.eti/nome-do-recurso/id-do-recurso |

Fonte: Autoria própria (2015)

Logo após a definição da estrutura e dos recursos deve-se, por meio de lógica de programação, filtrar as informações das requisições do *web service* abarcadas no *header* de cada requisição ao serviço. As informações contidas no *header* são de suma importância, pois, a partir delas que teremos as informações para tratar das requisições e enviar uma resposta. A Figura 2

mostra o exemplo de uma requisição pela URL www.myhost.com.br/events/1 onde temos como ação buscar dados do recurso Eventos através do método GET.


FIGURA 3 - *Header* de uma requisição GET, para buscar dados sobre eventos.
Fonte: Autoria própria (2015)

Para responder a essa requisição o serviço usa os *headers* para indicar um código de *status* para determinar sucesso ou fracasso da requisição, entre outros fatores. A Figura 3 mostra o *header* de resposta a requisição da Figura 2.

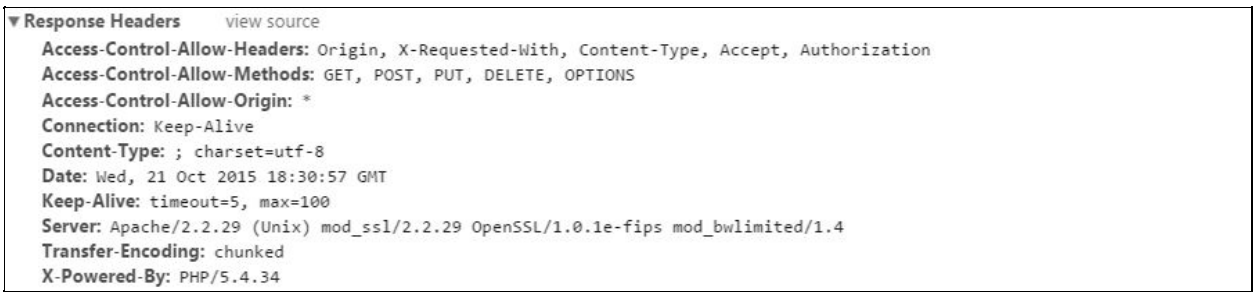


FIGURA 4 - *Header* de uma resposta a uma requisição GET, devolvendo dados do recurso Eventos.
Fonte: Autoria própria (2015)

Juntamente com o *header,* as respostas das requisições podem vir acompanhadas de um *body* onde o formato é definido no próprio *header* através do campo *Content-Type* setado como JSON como o resultado desta requisição que é mostrado na Figura 4.

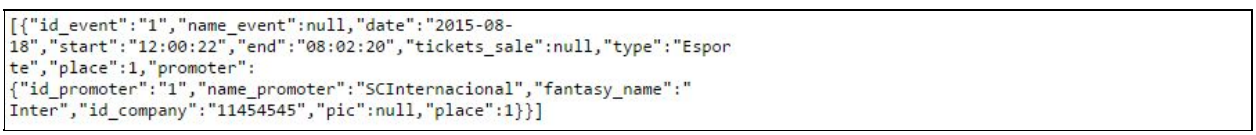


FIGURA 5 - *Body* de uma resposta a uma requisição GET. apresentando o contudo relacionado ao recurso Eventos.
Fonte: Autoria própria (2015)

Como citado na seção 2.4.4, umas das condições para que um serviço seja REST é necessário que o servidor não mantenha estados para comunicação com seus Clientes. Isso implica que para cada requisição do Cliente ao Servidor haja uma autenticação para que aquele usuário possa usar determinados recursos do serviço. Para que isso seja possível, foi implementado junto ao

serviço o protocolo OAuth 2 (Open Authentication 2), que permite uma autenticação padronizada para aplicações *web* e *mobile* através da utilização de tokens.

Para autenticar as aplicações Clientes e seus Usuários foi utilizado o padrão *User Credentials*. Onde o cliente usa o *client_id* e *client_secret* inseridos no *header* mais o *username* e *password* passados por meio de uma requisição POST. Estando autorizado o usuário recebe um *token* para ser sempre usado nas requisições posteriores. O cliente que tem a responsabilidade de armazenar o *token* para o usuário após ele ser gerado. O *token* tem validade curta por motivos de segurança a ataques em que o *token* possa ser descoberto, por isso junto com ele sempre vem o *refresh_token*, que renova as credenciais do usuário pelo caminho mais curto.A Figura 5, abaixo, demonstra o fluxo da autenticação do aplicativo.

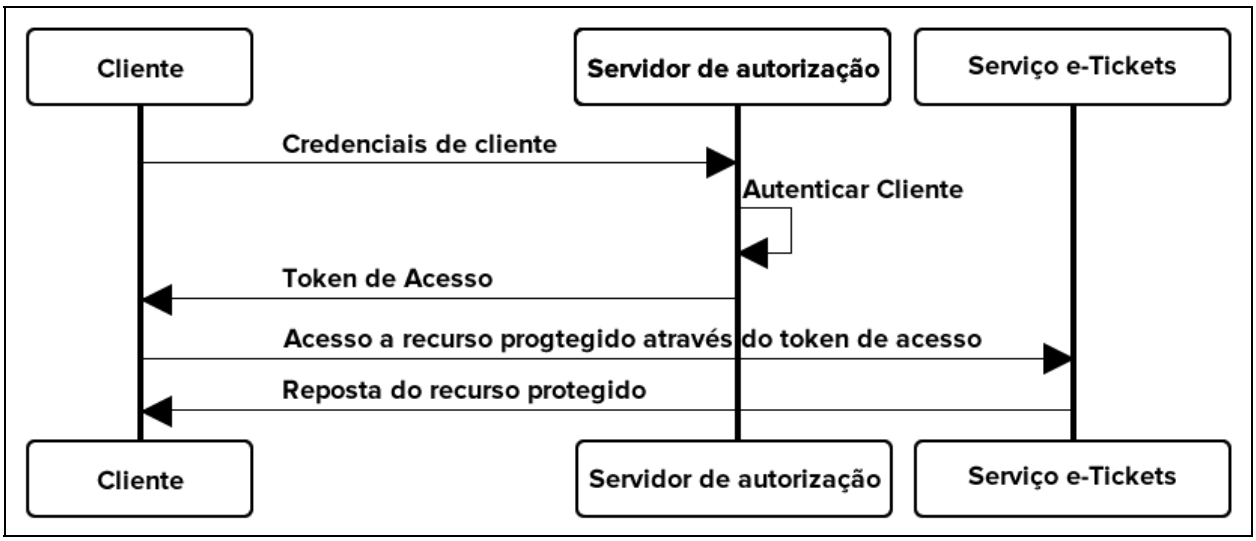


FIGURA 6 - Fluxo de autenticação *User Credentials*.
Fonte: *Oracle Docs*

### 3.3 Aplicativo

Esta seção apresentará a desenvolvimento da aplicação cliente do serviço e-Tickets, aplicativo que teve seu desenvolvimento baseado no *framework* para criação de aplicativos híbridos, o IONIC.

O aplicativo permite a um promoter que anuncie os seus eventos, possibilitando ao mesmo que defina todas as características pertencentes a este evento, como categorização, data, local, entre outros. Além disso, o aplicativo permite que um usuário possa efetuar a compra de ingressos para os eventos que estão anunciados, e foi implementado tendo em vista as necessidades de segurança inerentes a um aplicativo deste tipo.

### 3.3.1 Autenticação

Através da customização de um *service* da biblioteca AngularJs, é feita a passagem de informações entre o aplicativo e o *web service*, fazendo a autenticação do usuário e gerando as credenciais necessárias para o usuário utilizar os recursos disponíveis para o seu tipo. Estas credenciais são mantidas através de *Local Storage*, funcionalidade que o *web view* disponibiliza para o aplicativo.

A figura 7 abaixo ilustra uma tela da ferramenta, onde o usuário efetua a autenticação.


FIGURA 7 - Autenticação no aplicativo.
Fonte: Autoria própria (2015)

### 3.3.2 Usuários

Há dois tipos de usuários neste sistema, o usuário *Promoter* e o usuário normal. O primeiro representa uma instituição que lucra com a promoção de eventos públicos ou privados. O segundo poderá comprar *tickets* de qualquer evento público, e para os privados apenas se for convidado. Além disto, ele também poderá criar eventos públicos ou privados, mas não terá lucro com os mesmos.

O *Promoter* tem um papel importante, pois na descrição do evento constam suas informações e de sua marca. O sucesso de seu evento vai depender de como a sua marca é vista pelo público alvo.

### 3.3.3 Compras online

A figura 8 abaixo ilustra o fluxo de compra de ingressos compras online através da escolha de um evento, descrita nesta seção.


FIGURA 8 - Fluxo de compra de ingressos.
Fonte: Autoria própria (2015)

Por meio do uso de um *gateway* de pagamento *online* foi implementados um fluxo de compra de *tickets* para os eventos mantidos pelo sistema.

Para tanto, foi usada a API do Pagar.me. Pagar.me é uma empresa do ramo de gestão de pagamentos online que tem a maior taxa de conversão do mercado, assim, esta API foi utilizada pois, além de oferecer mais de uma forma de pagamento, também integra recursos que fazem parte do modelo de negócios do serviço. Aqui temos 3 partes envolvidas: o usuário (pagante), o *promoter* (recebedor 1) e o serviço (recebedor 2). Será cobrada uma taxa de serviço a cada transação de compra de ingressos, separando o pagamento em 90% para o *promoter* e 10% para os mantenedores do serviço. Para tanto, foi usada um função de *split-payment*, totalmente customizável com pagamento de recebedor automático, função que a API da Pagar.me desempenha com facilidade.

Para que o *Promoter* receba o seu pagamento ele deve ter seus dados cadastrados junto a Pagar.me. O usuário normal poderá ter uma conta ou digitar os dados na hora.

3.3.4 e-Tickets

Esta é a seção descreve o funcionamento de um e-Tickets. Toda vez que um evento for criado por um *promoter* ou por um usuário comum, uma chave privada será gerada e vinculada ao evento e seu criador. Toda vez que uma transação de compra de ingresso for finalizada, uma chave pública será gerada a partir da chave primária de seu evento. O código do *ticket* será criptografado usando essas chaves e o mesmo será mandado ao usuário já criptografado junto com a chave pública, assim, quando um usuário estiver no local do evento, basta que ele mostre o seu código criptografado junto com a sua chave pública para que o *promoter* possa descriptografar o código do ingresso usando a chave privada. A figura 9 abaixo demonstra este fluxo.

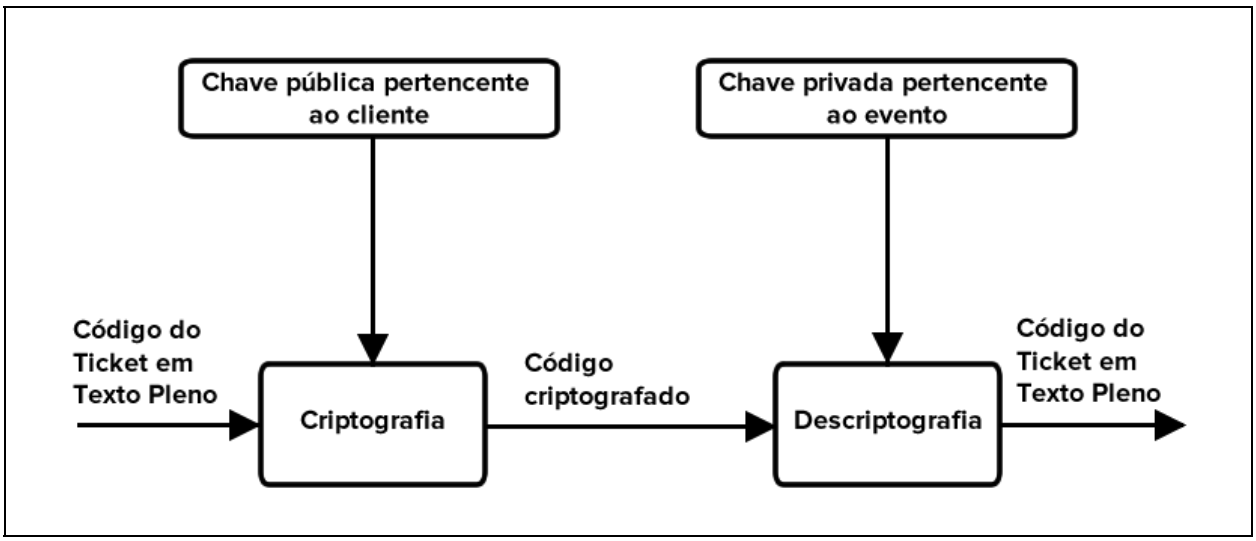


FIGURA 9- Fluxo da Criptografia assimétrica para o *ticket* eletrônico.
Fonte: Autoria própria (2015)

Para validar o ingresso basta que o usuário envie através do aplicativo o texto e a chave pública para que o *promoter* com a chave privada decodifique o código e verifique o mesmo junto ao sistema, conforme ilustrado na Figura 9. No aplicativo, esse fluxo é automático para ambos os usuários, bastando um toque de botão para que os dispositivos troquem tais informações**.**

## 4. Resultados Obtidos

Ao final deste projeto, concluiu-se o desenvolvimento da ferramenta e-Tickets. Para este projeto foram estimados resultados positivos para o ramo de Promoção e Divulgação de Eventos, com um ganho de visibilidade de mercado que será proporcional à popularização do aplicativo, por se tratar de uma ferramenta que oferece comodidade na hora de divulgar e comprar ingressos.

A ferramenta desenvolvida permite, de forma intuitiva e segura, que *promoters* de eventos façam a divulgação de seus eventos e a venda de ingressos para os mesmos, e também que usuários possam ter conhecimento destes eventos, e ainda fazer a compra dos ingressos para os mesmos.

Desta forma, o uso da ferramenta também proporciona como resultado uma maior comodidade em relação à maneira tradicional de se divulgar um evento, e também de se comercializar os ingressos para o mesmo.

## 5. Conclusão

Este artigo documentou o desenvolvimento da ferramenta e-Tickets, a qual permite a compra, venda e divulgação de ingressos, de forma online e segura, levando em consideração as tecnologias mais atuais para o tratamento destes tipos de transações. Conforme descrito, também foram utilizadas metodologias atuais para a análise e projeto da ferramenta.

A ferramenta desenvolvida consegue reunir três segmentos importantes quando trata-se deste nicho de mercado: a compra e venda de ingressos, a promoção e divulgação de eventos, e as compras online. Um de seus maiores diferenciais é a utilização do e-Tickets, pois o mesmo gera facilidade e comodidade tanto para o vendedor quanto para o comprador, abrindo um espaço ainda pouco explorado no mercado, apesar de ser muito concorrido.

Como trabalhos futuros, planeja-se tornar o serviço mais robusto em questões de segurança, estrutura, manipulação de recursos e documentação (apresentar a melhor forma de *design* e acessos ao métodos e recursos), pois os mesmos foram tratados de forma genérica no desenvolvimento da ferramenta, não sendo explorados todos os recursos de tais módulos. Além disso, pretende-se que a parte de *UX* (Experiência de usuário) seja aprimorada em futuras versões.

## Referências


AMBROS, L. **Diferença entre Aplicativos Nativos, Híbridos e Mobile Web Apps.** Disponível em:<http://www.luisaambros.com/blog/diferenca-entre-aplicativos-nativos-hibridos-e-mobile-web-apps/>. Acesso em 26 ago. 2015.

CESCA, C. G. G. **Organização de eventos:** manual para planejamento e execução. São Paulo: Summus, 1997.

CHEDE, C. **Desenvolvimento de apps – Parte 2: híbrido, nativo ou web?.** Disponível em: <https://www.ibm.com/developerworks/community/blogs/ctaurion/entry/desenvolvimento_de_apps-parte_2_hibrido_nativo_ou_web?lang=en, 2013> Acesso em 12 ago. 2015.

CHURCHILL, G; PETER, J. P. **Marketing: Criando Valor para os Clientes. 2 ed.** São Paulo: Saraiva 2003.

COELHO, L. C. **O papel da internet na economia.** Disponível em:<http://www.logisticadescomplicada.com/o-papel-da-internet-na-economia/ , 2011>. Acesso em: 11 ago. 2015.

DARLAN, DIEGO. **O QUE É PHP**. Disponível em: <http://www.oficinadanet.com.br/artigo/659/o_que_e_php, 2007>, Acesso em: 04 Ago. 2011.

FELIPINI, DAILTON. **Empreendedorismo na internet** - como agarrar esta nova oportunidade de negócios. LeBooks 2012.

FERREIRA, S. B. L.; CHAUVEL, M. A.; SILVEIRA, D. S. da. **Um estudo de usabilidade de sites de empresas virtuais**, v. 16, n. 2, p. 287-302, Maio/Ago. 2006. Disponível em:<http://www.scielo.br/pdf/prod/v16n2/08.pdf, 2006>Acesso em 08 ago. 2015.

FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. Tese (Doutorado em Informação e Ciência da Computação) – Universidade da Califórnia, Califórnia.

FISCHER, L. **Guia de introdução aos conceitos HTTP e REST.** Disponível em: <http://code.tutsplus.com/pt/tutorials/a-beginners-guide-to-http-and-rest--net-16340 , 2013>. Acesso em 06 jul. 2015.

GRIGORIK, I. **High-Performance Browser Networking.** O'Reilly, 2013.

GUERRATO, D. **Desenvolvimento ágil utilizando Scrum.**Disponível em: <http://tableless.com.br/desenvolvimento-agil-utilizando-scrum/>. Acesso em 08 ago 2015.

HOUAISS, A. **Míni Dicionário Houaiss da Língua Portuguesa**. Rio de Janeiro: Objetiva-Moderna, 2010.

MEIRELLES, G. F. **Tudo sobre eventos**. São Paulo: STS, 1999.

MENDES, L. Z. R. **E-commerce: origem, desenvolvimento e perspectivas.** Rio Grande do Sul**:** UFRGS, 2013**.**

RAVULAVARU, A. **Learning Ionic: Build real-time and hybrid mobile applications with Ionic.** Birmingham: Packt Publishing, 2015.

SCHWABER, K.; SUTHERLAND, J. **O guia do scrum**. Scrum Inc 2013.

SCHWABER, K. Guia do Scrum. Scrum Alliance, 2009.

SEBRAE. **O que você precisa saber sobre comércio eletrônico.** Disponível em: <http://www.sebrae.com.br/sites/PortalSebrae/sebraeaz/O-que-você-precisa-saber-sobre-comércio-eletrônico, 2015>. Acesso em 13 jun. 2015.

SILVA, L. A.; JACOB, D. A. **Comércio Eletrônico:** Atração e conquista de e-consumidores. Goiás: FANAP, 2011. Disponível em:<http://www.fanap.br/interlink/comercio_eletronico.pdf>. Acesso em: 04 ago. 2015.

VASCONELLOS, L. **Apps Híbridas com Cordova e Ionic.** Disponível em: <http://luisvasconcellos.com/2015/04/06/apps-hibridas-com-cordova-e-ionic.html ,2015>. Acesso em 08 mai. 2015.

W3C. **Web Services Architecture**: W3C Working Group 2004. Disponível em: <http://www.w3.org/TR/ws-arch/>. Acesso em: 13 mai. 2015.