

Aplicações Ricas para Web: Uso do GWT no Desenvolvimento de Controle de Reservas

Rafael Dani da Cunha¹, Sérgio M. Trad Júnior¹

¹Departamento de Ciência da Computação – Universidade Presidente Antônio Carlos (UNIPAC)
Rua Palma Bageto Viol S/N – Barbacena – MG – Brasil

rafael.danicunha@gmail.com, smtrad@gmail.com

Abstract. *The developing of Rich Internet Applications (RIA) are among of the new technologies available for developing web applications. These applications are similar to desktop applications, offering an rich, simple and intuitive interface for users. Among these technologies highlights the Google Web Toolkit (GWT). This article presents and compares the latest technologies for developing of rich internet applications, and also shows the development of a control application reserves academic resources to demonstrating the practical use of the Google Web Toolkit (GWT).*

Resumo. *O desenvolvimento de aplicações ricas para Internet (RIA) está entre as novas tecnologias disponíveis para a criação de aplicativos web. Estas aplicações assemelham-se com software desktop, visando um manuseio fácil e intuitivo dos usuários com uma interface mais rica do que as aplicativos web tradicionais. Este artigo apresenta e compara as principais tecnologias para o desenvolvimento de aplicações RIA e mostra o desenvolvimento de um software de controle de reservas de recursos acadêmicos com o propósito de demonstrar, de forma prática, o uso do Google Web Toolkit.*

1. Introdução

A Internet tem revolucionado o mundo dos computadores e das comunicações como nenhuma invenção foi capaz de fazer antes [Leiner et al. 2003]. Com o lançamento do navegador Mosaic¹ em 1993, que implementava suporte HTTP (Hypertext Transfer Protocol - Protocolo de Transferência de Hipertexto) e processamento de páginas HTML (HyperText Markup Language - Linguagem de Marcação de Hipertexto), não se imaginava o quanto a Internet ainda tinha para evoluir. Nessa época, o conteúdo era estático e as informações eram pouco usuais e relevantes [Bridee 2007]. Hoje a Internet tornou-se uma plataforma sólida na qual aplicações são desenvolvidas, proporcionando aos usuários novas experiências [Loosey 2006].

Com o aumento da demanda de aplicações web de maior complexidade, fez com que os programas atuais fossem além das possibilidades de uma aplicativo Web tradicional, surgindo então, as Aplicações Ricas para Internet (RIAs), que são software que se assemelham com programas desktop, ou seja, rodam na plataforma Web, mas possuem grande capacidade de interação com o usuário final. RIA faz com que o usuário tenha uma

¹Mosaic: Foi o primeiro navegador gráfico desenvolvido originalmente para roda no Unix, mas, logo depois, disponibilizado para *Macintosh* e *Windows* [Andreessen 2000].

experiência mais agradável e eficiente no uso de suas aplicações, isso é possível graças às tecnologias disponíveis [Gomes 2009].

Existem diversas tecnologias e ferramentas para auxiliar o desenvolvimento de aplicações RIAs, entre elas: Flex, JavaFX, Silverlight e Google Web Toolkit(GWT), onde cada qual descrita descrita na seção 3.2. Tecnologias e ferramentas para desenvolvimento de RIA.

1.1. Objetivos

Este trabalho tem como objetivo geral discutir os benefícios das tecnologias utilizadas para o desenvolvimento de aplicações RIAs, apresentando as vantagens e as desvantagens de cada uma. Especificamente, este trabalho objetiva desenvolver uma aplicação de controle de reservas de recursos acadêmicos com o propósito de demonstrar, de forma prática, o uso do Google Web Toolkit, e de beneficiar as instituições acadêmicas.

A aplicação irá permitir controlar, de forma prática e fácil, os recursos a serem disponibilizados para um melhor proveito das atividades acadêmicas. Possuindo, como requisitos funcionais, manter equipamentos, manter usuários, manter reservas, bloquear recursos, bloquear usuários e, como requisitos não funcionais, controle de acesso e uma interface amigável.

1.2. Motivação

Atualmente, criar aplicativos para web pode ser um processo longo e cansativo com alta incidência de erros. Os desenvolvedores podem passar grande parte do projeto trabalhando para contornar peculiaridades dos navegadores. Além disso, a criação, a reutilização e a manutenção de grandes bases de código JavaScript² e componentes AJAX pode ser difícil e delicada [Developers 2012].

A utilização do framework de código fonte aberto Google Web Toolkit (GWT) permite que você crie front end AJAX, utilizando a linguagem de programação Java, onde o compilador GWT se responsabiliza pela geração do código JavaScript correspondente, de forma otimizada e adaptada para que funcione automaticamente nos principais navegadores. Além disso, possibilita a depuração do código Java, de maneira produtiva, ou seja, linha a linha [Developers 2012].

Ainda pouco difundida entre os desenvolvedores, existem poucas aplicações desenvolvidas com o GWT, que ajudem a comprovar seus benefícios.

1.3. Metodologia

O estudo será desenvolvido por meio de bibliográficas, de levantamento de publicações técnicas especializadas e por meio de boas práticas de programação orientada a objetos.

2. Aplicações Baseadas na Web

Uma aplicação baseada na web é como um software acessível por um *thin-client*³ por intermédio de uma rede. Sua arquitetura mais comum é de três camadas, conforme Figura

²JavaScript: é uma linguagem de script criada por Brendan Ech, quando estava trabalhando na Netscape [Smeets et al. 2009].

³Thin-client: São aplicativos baseados em navegador, ou seja, são executados de dentro de um navegador [Smeets et al. 2009].

1. A primeira camada é representada pelo navegador. O lado servidor (*server side*), que apresenta componentes em diversas tecnologias tais como CGI, PHP, Java Servlet, juntamente com os componentes que interagem com o banco de dados, constitui a camada intermediária. A terceira e última camadas correspondem ao SGBD, que é responsável pelo gerenciamento e pelo armazenamento dos dados da aplicação [Alalfi et al. 2009].

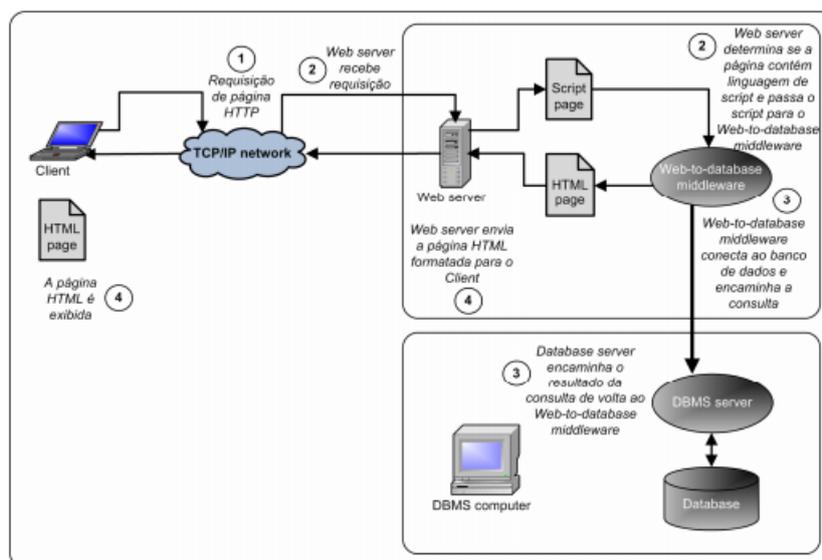


Figura 1. Componentes de uma aplicação baseada na web [Rob et al. 2009].

Aplicações tradicionais, baseadas na web, geram páginas que consistem em diferentes tipos de informações, tais como textos, imagens e formulários, podendo o seu conteúdo ser estático ou dinâmico. Conteúdos estáticos são providos por um servidor web, contendo apenas código em HTML e códigos executáveis no lado cliente (por exemplo, JavaScript). Conteúdos dinâmicos são providos pelo servidor da aplicação, resultado de execuções de vários scripts e de componentes [Gomes 2009].

3. Aplicações Ricas para Web (RIAs)

RIA (Rich Internet Application) é a revolução das aplicações Web, é mais que uma tecnologia, é um conceito. É o uso da Internet e das tecnologias disponíveis para criar uma experiência de uso para aplicações mais intuitivas e eficientes para o usuário. É a combinação de interatividade e funcionalidade do Desktop com a abrangência e flexibilidade da Web. Em aplicações RIA, tudo que é interface deve ser processado no cliente e o que for lógica de negócio deve ser processado no servidor [Bridee 2007].

RIA é mais um passo para o processo evolutivo da Internet. Essas aplicações se assemelham as aplicações desktop, fazendo com que seu uso seja mais fácil, oferecendo uma interface mais rica aos usuários [Loosey 2006].

Podemos citar algumas características destas aplicações, como: A aplicação é iniciada a partir de um navegador, incluindo o seu conteúdo. Os usuários que interagem com a aplicação, obtêm uma reação imediata da aplicação quando são solicitados novos

dados, ou seja, não há necessidade de renderizar⁴ a tela inteira [Gomes 2009].

As aplicações utilizam alguns controles de interface de usuário mais modernos como menu, painel de abas, entre outros elementos gráficos que não são utilizados em aplicações Web tradicionais. Permite o uso de operações comuns em aplicações Desktop, como *Drag & Drop* (arrasta e soltar), redimensionar, uso de animações, entre outras [Bridee 2007].

3.1. Vantagens e Desvantagens do RIAs

Como qualquer outro padrão de desenvolvimento de software, RIAs também possuem as suas vantagens e desvantagens, em comparação com os demais padrões.

3.1.1. Vantagens trazidas pela RIA

Além de benefícios internos, ao se criar uma aplicação rica para a Internet, podemos também citar a lista, a seguir, de benefícios adicionais [Smeets et al. 2009]:

- **Não necessita de instalação:** a aplicação é baixada automaticamente e roda dentro do navegador. O software que verdadeiramente roda a aplicação já está instalada na máquina do cliente;
- **Atualizações são automáticas:** novas versões da aplicação são baixadas automaticamente, toda vez que se visita a página da aplicação na rede;
- **Independência de plataforma:** uma aplicação rica na internet pode, potencialmente, rodar em qualquer plataforma e sistema operacional;
- **Maior Segurança:** as aplicações rodam em um ambiente restrito de um navegador web e são, portanto, menos sujeitas a causar danos do que aplicações que precisam ser instaladas;
- **Melhor resposta:** pois nem todas as ações do usuário requerem comunicação com o servidor, aplicações ricas na Internet tendem a reagir mais rápido que as clássicas aplicações da Internet;
- **Mais evolutiva:** uma grande parte do trabalho computacional, bem como a manutenção do estado do processo, pode ser descarregado ao cliente, de forma que o servidor possa tratar de muito mais usuários. E sem precisar manter estados ou pelo menos não tantos;
- **Maior eficiência na rede:** em clássicas aplicações da Internet, toda ação de usuário necessita que o servidor gere, novamente, a página completa e mande-a através de rede. No caso de aplicações de Internet rica, toda a UI da aplicação só precisa ser comunicada uma vez. Todos os demais pedidos ao servidor só necessitam que os dados correntes sejam enviados ao cliente.

3.1.2. Desvantagens da RIA

Existem também desvantagens em relação as RIA. A seguir algumas das desvantagens [Smeets et al. 2009]:

⁴Renderizar: É quando uma página já está aberta mais ainda não está pronta para ser apresentada ao usuário [Gomes 2009].

- **Necessita do JavaScript ou plug-in específico:** a aplicação completa roda por meio do interpretador JavaScript no cliente, o que sucede quando o cliente desliga completamente o JavaScript. Normalmente, a aplicação faz pouca coisa ou não faz nada. Ter um plano de backup para esses usuários é uma coisa óbvia, mas aí então você terá de manter duas aplicações separadas, o que está longe do ideal;
- **Não tem acesso a recursos do sistema:** como as aplicações rodam dentro de um navegador; elas estão limitadas quanto aos recursos que elas podem acessar. Por exemplo, uma aplicação Ajax não pode acessar o sistema de arquivos do cliente;
- **Difícil para os engines de pesquisa atingir indexação plena:** devido ao que a maioria dos *engines* de consulta ainda não suporta aplicações que fazem atualizações parciais de páginas ou usam *plug-ins* específicos, como o Flash, a maioria das aplicações de Internet ricas são pobremente indexadas por engines de busca. Os maiores engines de busca planejam melhorar o suporte para este tipo de aplicações à medida que a sua popularidade aumenta, mas sempre será difícil indexá-las o suficiente;
- **Problemas de Acessibilidade:** atualizações parciais de páginas usando JavaScript ou um plug-in específico podem quebrar a acessibilidade. O maior e mais conhecido problema é que as leitoras de telas existentes não manejam isso corretamente. Ainda que as leitoras de telas tentem fornecer um suporte melhor, desenvolvedores de aplicações deveriam sempre ter em mente, quando forem desenvolver aplicações, e ainda mais quando forem do tipo rica na Internet, porque acessibilidade é fácil de ser rompida;
- **Dependência de conexão Internet:** porque essas aplicações são servidas pela Web e rodam em um navegador, elas necessitam, pelo menos, de uma conexão inicial com a Internet. Mas, mesmo durante o uso, uma comunicação com a Internet é necessária para se comunicar com o servidor. Quando a conexão (temporariamente) não está disponível, as maiorias das aplicações ricas da Internet falham, não funcionam. Entretanto, há tentativas para se usar os serviços locais fornecidos pelo navegador para um armazenamento temporário enquanto uma conexão não se encontra disponível.

3.2. Tecnologias e ferramentas para desenvolvimento de RIA

Atualmente, existem diversas tecnologias e ferramentas, tais como, Flex, FavaFx, Silverlight, GWT, disponíveis no mercado que provem uma série de funcionalidades que o conceito RIA propõe, proporcionado uma experiência mais agradável ao usuário e uma interface mais rica como se verifica a seguir.

3.2.1. Flex

Conjunto de ferramentas para o desenvolvimento de aplicações ricas para Web, oferecendo ao usuário uma melhor interatividade e agilidade, comparada ao que poderia se obter utilizando HTML, utilizado da plataforma proprietária Adobe Flash. Desenvolver aplicações ricas em Flash é desafiador, por está voltada para designers, desenvolvimento utilizando timeline, fica estranho para o conceito de aplicações que não são movidas pelo tempo, mas por ações de seus operadores. Flex que foi criado, baseado no Flash, inicialmente tenha sido desenvolvido, pela Macromedia, como um produto comercial, teve

partes do código fonte liberado. Veio remover aquelas barreiras de entrada, fornecendo uma forma programática de desenvolver RIAs [Smeets et al. 2009].

Ao se desenvolver aplicações ricas para Web, utilizando Flex, está utilizando ferramentas fornecidas pela Adobe. Com isso, o Flex já vem com várias funcionalidades, como widgets, arrastar e soltar, gráficos vetoriais e animações, assim agilizado o desenvolvimento da aplicação.

A vantagem da utilização do Flex é que você desenvolve aplicações para rodar em ambientes controlados pelo plug-in do Flash. Mas, por outro lado, o plug-in pode não estar instalado, impedindo assim a execução de aplicação.

O Adobe Flex, depois do Adobe Air é uma boa opção para desenvolvimento de aplicações de internet rica, mesmo sendo grande parte dele um código-fonte aberto, ele necessita que os usuários instalem seus plug-ins proprietários antes de usar suas aplicações desenvolvidas em Flex [Smeets et al. 2009].

As aplicações desenvolvidas em Flex rodam sobre o Adobe Player 9, permitindo que os desenvolvedores estendam todas as suas funcionalidades, criando aplicações mais robustas e integradas com arquiteturas server-side, além de deixar o sistema com uma interface atraente e segura.

3.2.2. JavaFX

Possui uma linguagem de script para a fácil criação de mídia rica, com conteúdo interativo. No entanto está embutida na plataforma de desenvolvimento Java, portanto para implantar aplicações JavaFX na Web, está na realidade, implantando um applet Java, porém escrita com JavaFX. Um dos principais benefícios de um applet Java é um programa rodando no lado cliente, mesmo existindo algumas restrições de segurança quando se roda a partir do navegador, você pode fazer o que quiser com a applet Java. Uma das principais características é a reutilização de muitas partes dos códigos, sendo eles móveis ou em desktop [Smeets et al. 2009].

JavaFX é uma plataforma de desenvolvimento de software multimídia desenvolvida pela Oracle, baseada em Java para a criação de RIAs. A versão atual permite a criação para desktop, browsers e para telefones celulares, para outros recursos tecnológicos como TV, vídeo game entre outros já estão sendo planejados [Oracle 2012].

A atual versão do JavaFX inclui os seguintes componentes:

- O JavaFX SDK: Compilador e ferramentas para JavaFX. Gráficos, Media Web e documentos de textos com formatação;
- NetBeans IDE para JavaFX - Com a ajuda de uma paleta do Netbeans JavaFX o processo vira somente arrasta e solta (drag-n-drop), efeitos, animações e exemplos. Para eclipse também existe um plugin chamado Kenai;

As ferramentas e os plugins para os programas de criação de aplicativos: Plugin para Adobe Photoshop, Adobe Ilustrador que pode exportar gráficos com o código de JavaFX, ferramentas para converter SVG gráfico em JavaFX Script. Os plugins geram códigos em JavaScript que preservam o layout e a estrutura dos gráficos. As animações e os efeitos gráficos estáticos podem ser importados de maneira fácil e os desenvolvedores

estão buscando a forma mais eficiente de criação de conteúdos expressivos em aplicações que apareçam com desktops, na Internet, e em dispositivos móveis. Eles precisam construir alta fidelidade de GUIs que funcionam sem problemas em vários navegadores da Web, sistemas operacionais e dispositivos, sem tem que reescrever suas aplicações para cada tela. Para cumprir este objetivo, os desenvolvedores precisam trabalhar eficientemente com membros da equipe, tais como designers gráficos e autores de mídia para trocar áudio, vídeo e outros ativos de mídia avançada [Oracle 2012].

3.2.3. Silverlight

Silverlight é uma plataforma similar ao Flash, restrita a um subconjunto do .NET framework e, portanto visando à criação de aplicações ricas para Web [Smeets et al. 2009]. Permite também a criação de aplicações que rodem fora do navegador em seu desktop.

Utiliza Extensible Application Markup Language (XAML), que visa a facilitar o desenvolvimento de interfaces com o usuário, como, por exemplo, controles, animações, gráfico entre outros. Permitindo utilizar código gerenciado ou linguagem dinâmica para a lógica da aplicação [Microsoft 2012].

Entre as características do Silverlight podemos destacar [Microsoft 2012]:

- Cross-Browser e Cross-Platform: Funciona em todos os browsers populares, incluindo Internet Explorer, Firefox, Safari e Chrome. Além dos sistemas operacionais Windows e OS;
- Permite criação de aplicativos para Windows Phone;
- Possui gráficos atraentes, permitindo ao usuário arrastar, girar e zoom diretamente no navegador;
- Permite ser executado no navegador Web ou ser configurado, para que o usuário execute diretamente em seu computador.

3.2.4. Google Web Toolkit (GWT)

É um conjunto de ferramentas de desenvolvimento para construção e para otimização de aplicativos complexos, baseados na web. Seu objetivo é permitir o desenvolvimento produtivo de aplicações Web de alto desempenho, sem que o desenvolvedor tenha que se preocupar com as peculiaridades de cada navegador como XMLHttpRequest⁵ e JavaScript. GWT é utilizado em diversos produtos do Google, como exemplos podemos citar o Google Wave e a nova versão do Google AdWords [Developers 2012].

O Google Web Toolkit é um framework de código-fonte aberto, que tem como objetivo retirar a responsabilidade dos componentes de interface com o usuário do servidor. A interface com o usuário é escrita na linguagem Java e o compilador GWT converte o código resultante para código em Javascript e HTML compatível com os navegadores. Dessa maneira, toda a funcionalidade da interface com o usuário é disponibilizada no cliente, minimizando a comunicação com o servidor [Mesbah and Deursen 2009].

⁵XMLHttpRequest(XHR): é uma API disponível em linguagens de script para navegadores web tais como JavaScript [Developers 2012].

O GWT foi subdividido em três partes [Smeets et al. 2009]:

- **Compilador de Java para JavaScript:** parte mais importante, responsável por fazer toda a conversão do código-fonte escrito utilizando a linguagem Java seja transcrito para JavaScript;
- **Biblioteca de emulação JRE:** o JavaScript possui uma linguagem muito diferenciada do Java, por isso a função do GWT em compilar o código escrito para o JavaScript, para que isso possa acontecer, é necessário emular os construtores e as classes do núcleo Java, sendo, assim, traduzidos para o JavaScript;
- **Biblioteca de geração de UI:** consiste em várias subpartes, praticamente todo o código base, fornecido pelo GWT com componentes UI e suporte RPC.

Algumas características do GWT [Smeets et al. 2009];

- **Widgets e Layouts:** níveis mais básicos consistem na criação e inclusão de Widgets e Layouts. Basicamente, os Widgets são reais, como botões e campos de texto assim como painéis que fornecem layouts básicos;
- **Comunicação com os servidores:** a maioria das aplicações GWT precisa de algum tipo de *back end* para poder buscar e armazenar dados e processar outras ações, com o suporte fornecido pelo GWT, com o mecanismo de RPC;
- **Suporte a testes:** a vantagem sobre o Ajax é que facilita testar o código que escreve, podendo reutilizar em casos diferenciados. O GWT possuiu suporte próprio para testes sobre a framework de testes JUnit.

3.3. Comparativo entre tecnologias

Na tabela 1, podemos visualizar um comparativo entre algumas características das tecnologias para o desenvolvimento de RIAs:

- **Multiplataforma:** capacidade da aplicação desenvolvida de funcionar independente do sistema operacional que está sendo executado;
- **Necessidade de Plug-in:** necessidade de instalação de módulos de extensão, com objetivo de prover algumas funcionalidades especiais ou específicas;
- **Open-Source:** software sem custo de licenças, é de código fonte aberto, permitindo, assim, que qualquer pessoa contribua para o desenvolvimento do mesmo;
- **Acesso a recursos do navegador:** capacidade de acessar recursos disponibilizados pelo navegador, sem a necessidade de utilizar recursos de terceiros.
- **Acesso a sistema de arquivos local:** capacidade da aplicação de acessar o sistema de arquivo da máquina na qual está sendo executada;
- **Suporte a debug:** é o processo que permite ao programador encontrar e reduzir defeitos na aplicação desenvolvida;
- **Quirks mode:** técnica usada para manter a compatibilidade das páginas da web com os diversos navegadores existente no mercado .

4. Controle de Reservas

O sistema, a ser elaborado, visa a criar um facilitador para que as universidades possam controlar as reservas de equipamentos para um melhor proveito das atividades acadêmicas.

	Flex	JavaFX	SilverLigth	GWT
Multiplataforma	Sim	Sim	Sim	Sim
Necessidade de Plug-in	Sim	Sim	Sim	Não
Open-Source	Parcialmente	Não	Não	Sim
Acesso a recursos do navegador	?	?	?	Sim
Acesso a sistema de arquivos local	Não	Sim	Não	Não
Suporte a debug	Sim	Sim	Sim	Sim
Quirks mode	?	?	?	Sim

Tabela 1. Tabela comparativa das tecnologias

4.1. Características

Após análise de como o sistema deveria se comportar, conclui-se que o mesmo deverá possuir as seguintes características.

- O usuário poderá realizar, de qualquer lugar que possua acesso à internet, operações tais como:
 - Reserva de equipamentos;
 - Consulta reservas realizadas pelo mesmo;
- Visualização de informações sempre atualizadas e instantâneas;
- Implantação de nível de acesso para usuários como:
 - Administrador: terá o controle total sobre o sistema, podendo gerenciar usuários, cadastrar e consultar equipamentos, fazer reservas, e consultar reservas realizadas.
 - Professor: poderá realizar as reservas e as consultar.

4.2. Padrão MVP

Para o desenvolvimento da aplicação, foi aplicado o padrão MVP que tem por finalidade separar a camada de apresentação das camadas de dados e das regras de negócios, conforme figura 2.

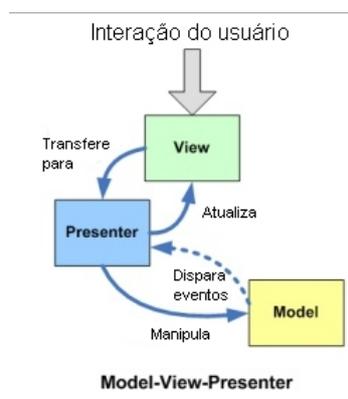


Figura 2. Padrão MVP

Assim podemos descrever cada camada de forma simplificada da seguinte maneira [Quicoli 2006]:

- **View:** é o formulário em que são exibidos os dados para o usuário, não possui regras de negócio a não ser disparar eventos que notifiquem mudanças de estado dos dados que são exibidos, assim como processamentos próprios. O objeto View implementa uma interface que tem por objetivos expor eventos e métodos que o Presenter necessite;
- **Model:** são os objetos que serão manipulados. Os objetos Model disponibilizam métodos que serão utilizados pelo Presenter, com o objetivo de atualizar valores do objeto, quando sofrerem alguma alteração na view;
- **Presenter:** é a ligação entre a View e Model; possui o papel de mediador entre eles. Ele que é o responsável de atualizar a view, quando o model é alterado e de sincronizar as informações contidas na view para o model.

4.3. Modelando a solução

Como finalidade de demonstrar a modelagem realizada, serão apresentados os Diagramas de Caso de Uso (figura 3) e o Diagrama de Classes (figura 4).

4.4. Diagrama de Casos de Uso

A figura 3 demonstra o Diagrama de Caso de Uso. que tem por finalidade fazer com que os sistemas, subsistemas e classes fiquem acessíveis e compreensíveis, por apresentarem uma visão externa sobre como os elementos podem ser utilizados no contexto [Booch et al. 2006].

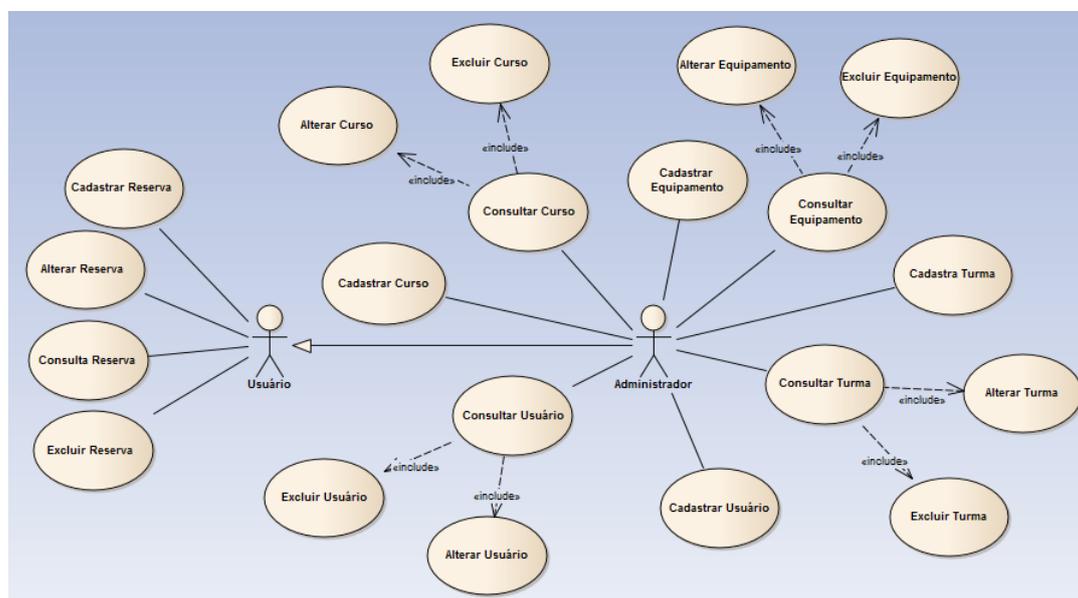


Figura 3. Diagrama de Caso de Uso

4.4.1. Caso de Uso: Cadastrar Curso

O sistema deverá permitir o cadastro de cursos existentes na instituição. As informações que irão compor o cadastro de cursos são:

- Descrição: Nome completo do curso.

- Orientador: Nome completo do funcionário responsável pelo curso.
- Situação: Podem ser:
 - Ativo
 - Inativo

Perfil de usuário:

- Administrador

Pré-condição:

- Não existir duplicidade de cadastro, ou seja, os valores de descrição e orientador iguais.

4.4.2. Caso de Uso: Cadastrar equipamentos

O sistema deverá permitir o cadastro de equipamentos. As informações que irão compor o cadastro de equipamentos são:

- Número de patrimônio: Número de identificação do equipamento na instituição.
- Descrição: Descrição para identificação do equipamento.
- Situação: Podem ser:
 - Ativo
 - Inativo

Perfil de usuário:

- Administrador

Pré-condição:

- Não existir duplicidade de cadastro, ou seja, número de patrimônio igual.

4.4.3. Caso de Uso: Cadastrar Turma

O sistema deverá permitir o cadastro de turmas. As informações que irão compor o cadastro de turmas são:

- Número da turma: Número de identificação da turma na instituição.
- Descrição: Descrição para identificação da turma.
- Curso: Curso no qual a turma está vinculada.
- Sala: Número da sala.

Perfil do usuário:

- Administrador

Pré-condição:

- Não existir duplicidade de cadastro, ou seja, número da turma igual.

4.4.4. Caso de Uso: Cadastrar Usuário

O sistema deverá permitir o cadastro de usuários. As informações que irão compor o cadastro de usuário são:

- Nome: Nome completo do usuário.
- Login: Nome de identificação para a utilização do sistema.
- Senha: Senha pessoal para utilização do sistema.
- E-mail. Email do usuário.
- Tipo de Usuário: Podem ser:
 - Administrador
 - Operador
- Situação: Podem ser:
 - Ativo
 - Inativo

Perfil de usuário:

- Administrador

Pré-condição:

- Não existir duplicidade de cadastro, ou seja, nomes, logins, e-mails iguais.

4.4.5. Caso de Uso: Consulta Curso

O sistema deverá permitir a consulta de cursos. Os critérios de pesquisa para consultar são:

- Situação [Ambos, Ativos, Inativos]: Opção padrão "Ativos". Critério obrigatório.
- Descrição: Mínimo de três letras.
- Orientador: Mínimo de três letras.

Após o preenchimento de um ou de mais critérios de pesquisa, o usuário, operador do sistema, poderá pressionar um botão que fará com que o sistema processe a pesquisa. Ao concluir o processamento da pesquisa, o sistema deverá exibir uma tabela que liste os cursos que atendam aos critérios.

Perfil de usuário:

- Administrador

4.4.6. Caso de Uso: Consultar Equipamento

O sistema deverá permitir a consulta de equipamentos. Os critérios de pesquisa para consultar são:

- Situação [Ambos, Ativos, Inativos]: Opção padrão "Ativos". Critério obrigatório.
- Descrição: Mínimo de três letras.

Após o preenchimento de um ou de mais critérios de pesquisa, o usuário, operador do sistema, poderá pressionar um botão que fará com que o sistema processe a pesquisa. Ao concluir o processamento da pesquisa, o sistema deverá exibir uma tabela que liste os equipamentos que atendam aos critérios.

Perfil de usuário:

- Administrador

4.4.7. Caso de Uso: Consultar Turma

O sistema deverá permitir a consulta de turmas. Os critérios de pesquisa para consultar são:

- Curso: Critério Obrigatório.
- Número da turma.
- Descrição: Mínimo de três letras.
- Sala.

Após o preenchimento de um ou de mais critérios de pesquisa, o usuário, operador do sistema, poderá pressionar um botão que fará com que o sistema processe a pesquisa. Ao concluir o processamento da pesquisa, o sistema deverá exibir uma tabela que liste as turmas que atendam aos critérios.

Perfil de usuário:

- Administrador

4.4.8. Caso de Uso: Consultar Usuário

O sistema deverá permitir a consulta de usuários. Os critérios de pesquisa para consultar são:

- Situação [Ambos, Ativos, Inativos]: Opção padrão "Ativos". Critério obrigatório.
- Nome: Mínimo de três letras.
- Login.
- Tipo de usuário [Ambos, Administrador, Operador].

Após o preenchimento de um ou de mais critérios de pesquisa, o usuário, operador do sistema, poderá pressionar um botão que fará com que o sistema processe a pesquisa. Ao concluir o processamento da pesquisa, o sistema deverá exibir uma tabela que liste os usuários que atendam aos critérios.

Perfil de usuário:

- Administrador

4.4.9. Caso de Uso: Editar Curso

O sistema deverá permitir a alteração de dados do curso, observando-se os mesmos critérios definidos no cadastro.

Perfil de usuário:

- Administrador

4.4.10. Caso de Uso: Editar Equipamento

O sistema deverá permitir a alteração de dados do equipamento, observando-se os mesmos critérios definidos no cadastro.

Perfil de usuário:

- Administrador

4.4.11. Caso de Uso: Editar Turma

O sistema deverá permitir a alteração de dados da turma, observando-se os mesmos critérios definidos no cadastro.

Perfil de usuário:

- Administrador

4.4.12. Caso de Uso: Editar Usuário

O sistema deverá permitir a alteração de dados do usuário, observando-se os mesmos critérios definidos no cadastro.

Perfil de usuário:

- Administrador

4.4.13. Caso de Uso: Excluir Curso

O sistema deverá permitir que um curso, já cadastrado, seja excluído, via tela de consulta.

Perfil de Usuário:

- Administrador

Pré-condição

- Não exista nenhum vínculo com outros registros do sistema.

4.4.14. Caso de Uso: Excluir Equipamento

O sistema deverá permitir que um equipamento, já cadastrado, seja excluído, via tela de consulta.

Perfil de usuário:

- Administrador

Pré-condição

- Não exista nenhum vínculo com outros registros do sistema.

4.4.15. Caso de Uso: Excluir Turma

O sistema deverá permitir que uma turma, já cadastrado, seja excluído, via tela de consulta.

Perfil de usuário:

- Administrador

Pré-condição

- Não exista nenhum vínculo com outros registros do sistema.

4.4.16. Caso de Uso: Excluir Usuário

O sistema deverá permitir que um usuário, já cadastrado, seja excluído, via tela de consulta.

Perfil de usuário:

- Administrador
- Pré-condição
- Não exista nenhum vínculo com outros registros do sistema.

4.4.17. Caso de Uso: Cadastrar Reserva

O sistema deverá permitir o cadastro de reservas. As informações que irão compor o cadastro de turmas são:

- Data da Reserva.
- Hora Inicial.
- Hora Final.
- Turma.
- Usuário. Usuário no qual realizou a reserva.
- Equipamentos[1..n].

Perfil do usuário:

- Administrador
- Operador

Pré-condição:

- Não existir duplicidade de cadastro, ou seja, data da reserva, hora inicial e turma iguais.
- Deve possuir no mínimo um equipamento.

4.4.18. Caso de Uso: Alterar Reserva

O sistema deverá permitir a alteração de dados da reserva, observando-se os mesmos critérios definidores no cadastro.

Perfil do usuário:

- Administrador
- Operador

4.4.19. Caso de Uso: Consultar Reserva

O sistema deverá permitir a consulta de reservas. O critério de pesquisa para consultar será:

- Data da Reserva. Critério obrigatório.

Após preenchimento de um ou mais critérios de pesquisa, o usuário, operador do sistema, poderá pressionar um botão que fará com que o sistema processe a pesquisa. Ao concluir o processamento da pesquisa, o sistema deverá exibir uma tabela que liste as reservas que atendam aos critérios.

Perfil do usuário:

- Administrador
- Operador

4.4.20. Caso de Uso: Excluir Reserva

O sistema deverá permitir que uma reserva, já cadastrada, seja excluído, via tela de consulta.

Perfil do usuário:

- Administrador
- Operador

4.5. Diagrama de Classe

A figura 4 apresenta o Diagrama de Classe que tem por objetivos mostrar um conjunto de classes, de interfaces e de colaboradores com os seus respectivos relacionamentos. Os Diagramas de Classe são de grande importância para a construção de sistemas executáveis por intermédio de engenharia de produção e reversa, assim como para a visualização, para a especificação e para a documentação dos modelos estruturais [Booch et al. 2006].

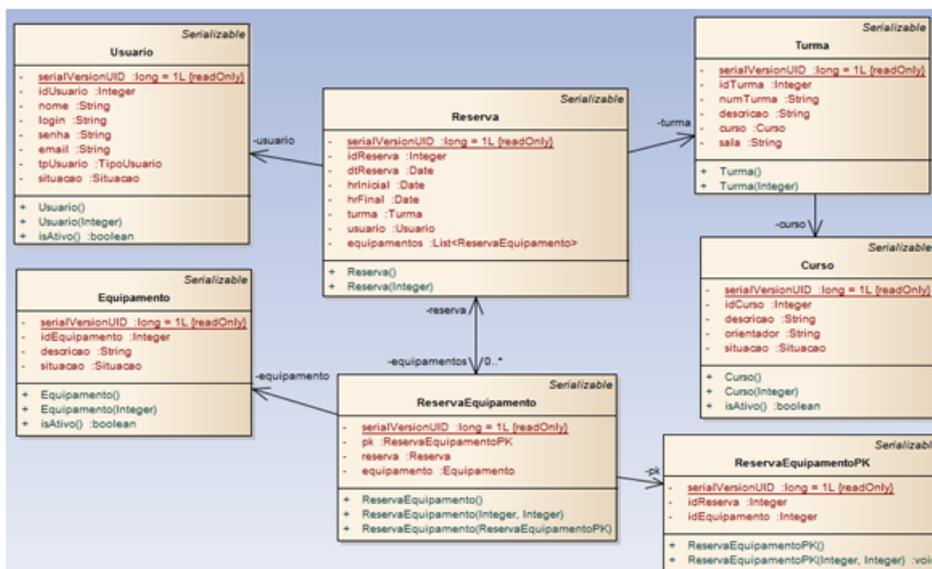


Figura 4. Diagrama de Classes de Domínio

4.6. Interfaces do Sistema

Para ilustrar o sistema que foi desenvolvido, serão apresentadas algumas telas como: cadastro de turma, cadastro de reservas, consulta de usuários, consulta de reservas.

A tela de "Cadastro de Turma", ilustrada na figura 5, permite que o usuário inclua uma nova turma, conforme descrito no caso de uso "Cadastrar Turma".

The screenshot shows a web application interface titled "Controle de Reservas". On the left, there is a sidebar menu with three main sections: "Cadastros Básico", "Consulta", and "Movimentação". Under "Cadastros Básico", there are links for "Curso", "Equipamento", "Turma", and "Usuário". Under "Consulta", there is a link for "Consulta". Under "Movimentação", there is a link for "Movimentação". The main content area is titled "Cadastrar Turma" and contains the following fields: "Número da turma" (with a sub-label "Descrição" above it), "Curso" (a dropdown menu), and "Sala". At the bottom of the form are two buttons: "Salvar" and "Cancelar". In the bottom right corner of the application window, there is a small text credit: "Desenvolvido por Rafael Dani da Cunha".

Figura 5. Cadastro de turma

A tela de "Cadastro de Reservas", ilustrada na figura 6, permite que o usuário inclua uma reserva, seguindo as regras descritas no caso de uso "Cadastrar Reserva".

The screenshot shows a web application interface titled "Controle de Reservas". On the left, there is a sidebar menu with three main sections: "Cadastros Básico", "Consulta", and "Movimentação". Under "Cadastros Básico", there are links for "Curso", "Equipamento", "Turma", and "Usuário". Under "Consulta", there are links for "Consulta", "Cadastrar Reserva", and "Consultar Reservas". Under "Movimentação", there is a link for "Movimentação". The main content area is titled "Cadastrar Reserva" and contains the following fields: "Data", "Horário (Início)", and "Horário (Fim)" (each in its own input box), and "Turma" (a dropdown menu). At the bottom of the form are two buttons: "Salvar" and "Cancelar". In the bottom right corner of the application window, there is a small text credit: "Desenvolvido por Rafael Dani da Cunha".

Figura 6. Cadastro de reservas

A tela de "Consulta de Usuários", ilustrada na figura 7, permite que o operador liste os usuários já cadastrados no sistema, conforme descrito no caso de uso "Consultar Usuário". Ainda permitindo que o mesmo altere ou delete o registro selecionado, conforme os casos de uso "Alterar Usuário" e "Deletar Usuário".

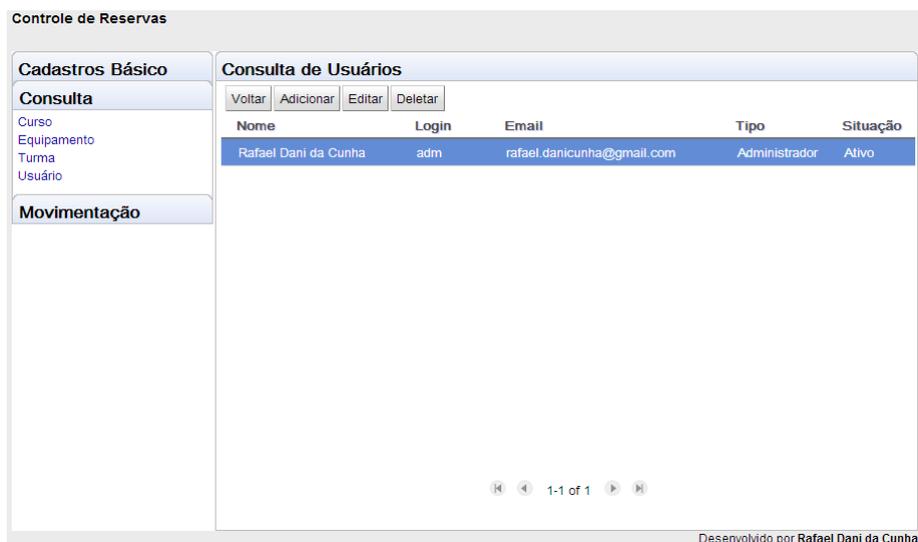


Figura 7. Consulta de usuários

A tela de "Consulta de Reserva", ilustrada na figura 8, permite que o usuário liste as reservas já cadastradas no sistema, conforme descrito no caso de uso "Consultar Reservas". Ainda permitindo que o mesmo altere ou delete o registro selecionado, conforme os casos de uso "Alterar Usuário" e "Deletar usuário".

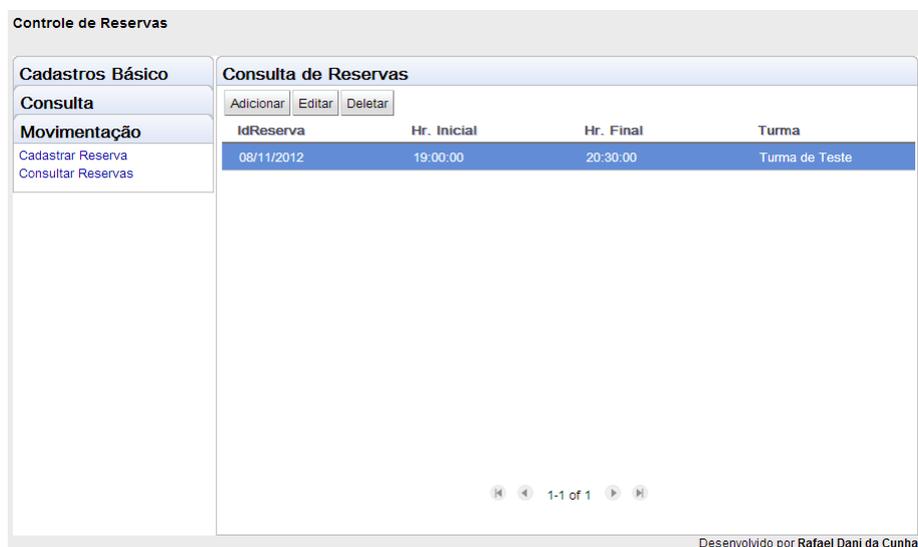


Figura 8. Consulta de reservas

5. Conclusão

Este trabalho teve como objetivo apresentar um comparativo entre algumas tecnologias disponíveis no mercado para criação de aplicações ricas, visando a ser referência na escolha de uma solução simples e barata para o desenvolvimento das mesmas, atendendo, assim, às expectativas do mercado.

Para alcançar o objetivo proposto, foram realizando estudos sobre as tecnologias Flex, JavaFX, Silverlight e Google Web Toolkit nos quais foi possível observar as vantagens e as desvantagens de cada uma sobre o processo de desenvolvimento de uma aplicação. Por intermédio da pesquisa e do desenvolvimento da aplicação para controle de reservas, foi possível perceber o grande potencial do Google Web Toolkit em relação às demais.

O Google Web Toolkit, além de ser uma tecnologia totalmente open-source que permite o desenvolvedor escrever os códigos na linguagem Java, que é amplamente conhecida e difundida no mercado, tirando, assim, a necessidade de aprender várias outras linguagens para alcançar o objetivo proposto. O GWT, por possuir um compilador no qual converter o código-fonte para outra linguagem, que seja interpretada pelo navegador, assim não sendo necessária a instalação de plug-in na máquina do cliente como as outras tecnologias citadas.

Existem outras tecnologias disponíveis no mercado, para o desenvolvimento de aplicações ricas que não foram contempladas neste trabalho, tal como JSF - JavaServer Faces, permitindo que a pesquisa seja estendida, incluindo vantagens e desvantagens em relação as tecnologias citadas.

Referências

- Alalfi, M. H., R., C. J., and R., D. T. (2009). *Modeling methods for web application verification and testing: State of the art*.
- Andreessen, M. (2000). Mosaic - the first global web browser. http://www.livinginternet.com/w/wi_mosaic.htm. Último acesso em: 02/05/2012.
- Booch, G., Rumbaugh, J., and Jacobson, I. (2006). *UML - Guia do Usuário*.
- Bridee, E. (2007). Web 2.0 & ria.
- Developers, G. (2012). Google web toolkit. <https://developers.google.com/web-toolkit/>. Último acesso em: 02/11/2012.
- Gomes, J. N. (2009). Richblocks - um framework para implantar interfaces ria em sistemas web.
- Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., Postel, J., Roberts, L. G., and Wolff, S. (2003). Brief history of the internet.
- Loosey, C. (2006). Rich internet applications: Design, measurement, and management challenges.
- Mesbah, A. and Deursen, A. V. (2009). An architectural style for ajax.
- Microsoft (2012). Silverlight overview. [http://msdn.microsoft.com/pt-br/library/bb404700\(v=vs.95\).aspx](http://msdn.microsoft.com/pt-br/library/bb404700(v=vs.95).aspx). Último acesso em: 25/09/2012.
- Oracle (2012). Javafx - the rich client platform. <http://www.oracle.com/technetwork/java/javafx/overview/index.html>. Último acesso em: 10/10/2012.
- Quicoli, P. (2006). O padrão mvp (model-view-presenter). <http://www.devmedia.com.br/o-padroao-mvp-model-view-presenter/3043>. Último acesso em: 25/10/2012.

Rob, P., Coronel, C., and Morris, S. (2009). *Database Systems: Design, Implementation, and Management*.

Smeets, B., Boness, U., and Bankras, R. (2009). *Programando Google Web Toolkit - Do Iniciante ao Profissional*.